递归论导论

● 郭世铭 著

现代逻辑丛书

DIGUI
LUN
DAOLUN



ISBN 7-5004-2295-4/B · 478

定价：11.00 元

现代逻辑丛书

●此项研究成果受国家社会科学基金资助●

递归论导论

郭世铭　著

中国社会科学出版社

（京）新登字 030 号

图书在版编目（CIP）数据

递归论导论/郭世铭著．—北京：中国社会科学出版社，1998.8

ISBN 7-5004-2295-4

Ⅰ．递…　Ⅱ．郭…　Ⅲ．递归论-概论　Ⅳ．O141.3

中国版本图书馆 CIP 数据核字（98）第 14107 号

中国社会科学出版社出版发行
（北京鼓楼西大街甲 158 号　邮编 100720）
北京新魏印刷厂印刷　新华书店经销
1998 年 8 月第 1 版　1998 年 8 月第 1 次印刷
开本：850×1168 毫米 1/32　印张：5.625　插页：2
字数：138 千字　印数：1—2000 册
定价：11.00 元

《现代逻辑丛书》出版说明

现代逻辑内容很丰富，特别是符号逻辑或称数理逻辑，包括几个分支，如：逻辑演算，集合论，模型论，递归论，证明论等。在古典逻辑演算以外，近年来模态逻辑有了很大的发展，它又被称作哲学逻辑。

符号逻辑不仅内容丰富，还和许多学科如哲学、数学、计算机科学、语言学及心理学等有联系，影响及于这些学科，有些影响甚至是带根本性的。

我国大学的逻辑专业，计算机专业，数学专业，哲学专业等，都开设和符号逻辑有关的课程。

但是，这方面介绍性的书籍和教材在国内还不多见。本丛书的目的是提供一批叙述简明易懂和不需要较多数学知识的入门性书籍和教材。

《现代逻辑丛书》被列入国家第七个五年计划期间重点研究课题，由北京大学哲学系逻辑教研室王宪钧教授主编，教研室及校外任课教员执笔编写。

目　　录

第一章　导言和预备知识……3
§1.1　算法……3
§1.2　部分函数……6
第二章　图灵可计算函数……9
§2.1　图灵机的基本概念……10
§2.2　图灵可计算函数……15
§2.3　正则的图灵机……20
§2.4　可计算性与函数运算……30
习　题……36
第三章　部分递归函数……39
§3.1　原始递归函数……39
§3.2　部分递归函数……48
习　题……51
第四章　通用函数和通用图灵机……55
§4.1　图灵机的算术化……55
§4.1.1　图灵机的编号……55
§4.1.2　瞬时描述的编号……58
§4.1.3　图灵机的算术化……61
§4.2　通用函数和通用图灵机……63
§4.3　对角线方法……65
习　题……68
第五章　s-m-n 定理和递归定理……69
§5.1　丘奇论题……69

§5.2 s-m-n 定理 …… 72
§5.3 递归定理 …… 76
习 题 …… 77
第六章 哥德尔不完全性定理 …… 79
§6.1 形式算术 …… 79
§6.2 PA 系统的算术化 …… 84
§6.3 数字可表示性 …… 87
§6.4 哥德尔不完全性定理 …… 93
第七章 递归可枚举集 …… 101
§7.1 判定问题 …… 101
§7.2 递归可枚举集 …… 104
习 题 …… 115
第八章 m-归约和 m-度 …… 116
§8.1 m-归约 …… 116
§8.2 指标集 …… 121
§8.3 产生集、创造集和单纯集 …… 127
习 题 …… 133
第九章 图灵度 …… 134
§9.1 相对可计算性 …… 134
§9.2 图灵度 …… 143
§9.3 极限 …… 148
习 题 …… 151
第十章 波斯特问题 …… 153
§10.1 波斯特问题和单纯集 …… 153
§10.2 优先方法 …… 159
§10.3 穆契尼克和弗里德伯格的证明 …… 165
习 题 …… 171
索引 …… 173
参考书目 …… 178

第一章

导言和预备知识

递归论是数理逻辑的一个重要分支，它与证明论、模型论及公理集合论合称“四论”，构成数理逻辑的核心内容。递归论也叫**可计算性理论**，它所研究的对象是递归函数（也叫可计算函数）及其定义域。对于递归函数有许多不同（但彼此等价）的定义方法，这些定义的目的都是为了刻画直观的“算法”概念，因此我们的讨论也就从算法入手。

§1.1 算　法

算法(algorism)也叫**能行方法**(effective method)或**能行过程**(effective procedure)，是一个相当古老的概念，其历史可以上溯到古希腊时代。著名的阿基米德辗转相除法和奥拉斯托散纳筛法就是古代算法的两个杰出范例。我们不妨以它们为例分析一下算法的基本特征。

例 1(阿基米德辗转相除法)　任给两个正整数 m 和 n，可以按如下方法求得它们的最大公约数 (m,n)：

(1) 先用其中一个数去除另一个数，比如用 n 去除 m。如果能除尽，除数 n 就是最大公约数；如果除不尽，有余数 $r_1(0<r_1<n)$，进入下一步；

(2) 用上一步所得的余数(即 r_1)去除上一步的除数(即 n)。如果能除尽，则这次的除数 r_1 就是最大公约数；如果除不尽，有余数

$r_2(0<r_2<r_1)$，则再重复本条的过程。

由于每除一次余数都会减小，因此在有穷次除法之后总可以使余数成为0(除尽)。这最后一次除法的除数就是我们所要找的最大公约数(m,n)。比如我们要计算75和27的最大公约数(75,27)，可以如下进行：

① 先用27去除75，商2余21(即$r_1=21$)；

② 再用21去除27，商1余6(即$r_2=6$)；

③ 再用6去除21，商3余3(即$r_3=3$)；

④ 再用3去除6，商2余0(除尽)。

于是(75,27)=3。

例2(奥拉斯托散纳筛法) 任给正整数m，可以按如下方法制造m以内的素数表：

(1) 依次写下从2到m的全体正整数；

(2) 在第一个数2上画一个圈(以示保留)，然后在2的其他倍数上画×(以示删除)；

(3) 在未作标记(圈或×)的最小的数上画圈，并在该数的其他倍数上画×；反复进行这一过程，直到所有的数上都有标记为止。这时，画圈的数就是m以内的全部素数。

比如我们要作10以内的素数表，其具体过程如下：

(1) 先写下从2到10的全部正整数：

2　3　4　5　6　7　8　9　10

(2) 圈住2，并在2的其他倍数上打×，得到

②　3　~~4~~　5　~~6~~　7　~~8~~　9　~~10~~

(3) 圈住不带标记的最小数——3，并在3的其他倍数上打×，得到

②　③　~~4~~　5　~~6~~　7　~~8~~　~~9~~　~~10~~

(4) 圈住不带标记的最小数——5，并在5的其他倍数上打×，得到

②　③　~~4~~　⑤　~~6~~　7　~~8~~　~~9~~　~~10~~

(5) 圈住不带标记的最小数——7。至此所有的数上都有了标记，过程结束。我们得到10以内的素数表：

2 3 5 7。

从上面两个例子可以看出算法的基本特征是：

1. 所有动作都严格按照事先给定的(有穷多条)规则进行；

2. 每一步究竟按哪一条规则行动也完全是确定的——或者由规则直接确定，或者由规则和上一步的结果确定；

3. 整个过程在有穷步内结束。

如果用一句更通俗的话来说，所谓算法就是一种死办法。利用算法解题并不需要特别的聪明才智，只要严格地照章办事就行，无论是谁来做，其过程和结果都完全相同。

利用算法所解的问题，大致可以分为两类：一是计算函数，二是判定是非。对于一个函数，如果我们能找到一个计算它的算法，就称该函数是**能行可计算**的或**算法可计算**的。

所谓判定是非的问题，是确定某一类对象(比如自然数)中的任何一个(或几个)是否具有某种性质(或某种关系)。如果我们能找到一个算法来解决某个判定问题，就称该问题是**能行可判定**的或**算法可判定**的。

例如，"一个任给的自然数是否素数"、"一对任给的自然数是否互素"、"一个任给的整系数一元二次方程是否有整数根"都是能行可判定的问题。

自古以来，算法就是一个重要的研究课题。直到本世纪初年以前，人们一直满足于直观的算法概念，那时，人们普遍认为(可能并不完全自觉)，对于有关自然数的种种问题，这类算法总是有的，研究者的任务就是把它们实际找出来。1900年希尔伯特所提出的二十三个著名的问题中，就有一个判定问题——第十问题。希尔伯特对该问题的陈述是：

给出一个过程来确定任意的整系数刁番图方程是否

有(整数)解。[①]

希尔伯特是直截了当地要求给出这样一个算法，而不是要求证明存在这种算法。这表明在那个时代这类算法的存在性并未受到怀疑。(当然，现在我们已经知道希尔伯特所要求的算法并不存在。)

但是，此后三十多年的发展却使人们改变了看法。人们逐渐感到所期待的某些算法(如谓词演算的判定算法)可能并不存在。但是，要想证明一个算法不存在，就不能再满足于算法的直观概念，而需要对它作严格的数学刻画了。在二十世纪三十年代，波斯特、哥德尔、丘奇、图灵、马尔科夫等人分别从不同的角度对算法概念作了严格的刻画，他们所定义的函数类，就是本书所要讨论的基本对象。

§1.2 部分函数

尽管我们也时常说及实数的计算乃至复数的计算，但我们实际上所能做的只是对有理数、整数、自然数的计算。而对有理数的计算可以归结为对整数的计算加上一个确定小数点位置的定位法，而对整数的计算又可以归结为对自然数的计算加上一个确定正负号的符号法则。因此实际的计算问题的核心是对自然数的计算。

今后本书中所讲的"数"都是指自然数，"集合"都是指自然数集合，"谓词"("关系")都是指自然数上的谓词(关系)，"函数"也都是指自然数中的函数。我们用 N 表示全体自然数所组成的集合(全集)，即

$$N=\{0, 1, 2, \cdots\}$$[②]

---

① 所谓刁番图方程指的是只讨论其整数解的整系数多项式方程。

② 与某些数学分支不同，在递归论(乃至整个数理逻辑)中，都是把0作为自然数的。

以后，我们用小写的拉丁字母 $x, y, z, \cdots$ 表示在 N 上取值的变元，用大写的拉丁字母 $A, B, C, \cdots$ 表示自然数集或自然数 n 元组的集合；用大写的拉丁字母 $P, Q, R, \cdots$ 表示自然数上的 n 元谓词（关系）。

在讨论递归函数时，我们不仅要考虑对全体自然数（或自然数的全体 n 元组）都有定义的那些函数（全函数），也要考虑只对一部分自然数（一部分 n 元组）有定义的函数。

定义 1.2.1 设 n 为一个任意固定的正整数，$A \subseteq N^n$。从 A 到 N 中的映射 f

$$f: A \to N$$

称为 n 元**部分函数**，简称**函数**。如果 $A = N^n$，则称 f 为 n 元**全函数**。f 的定义域记为 $\mathrm{dom}\, f$，f 的值域记作 $\mathrm{ran}\, f$。

例如自然数的加法、乘法都是二元的全函数，而减法和除法则都只是部分函数而不是全函数。

定义 1.2.2 设 n 为任意固定的正整数，P 是自然数上的 n 元谓词，集合

$$\{(x_1, \cdots, x_n) \mid P(x_1, \cdots, x_n) \text{为真}\}$$

称为谓词 P 的**外延集**。

定义 1.2.3 设 n 为任意固定的正整数，$A \subseteq N^n$ 是 n 元组的集合，函数

$$C_A(x_1, \cdots, x_n) = \begin{cases} 1 & \text{当}(x_1, \cdots, x_n) \in A, \\ 0 & \text{当}(x_1, \cdots, x_n) \notin A。\end{cases}$$

称为 n 元组集合 A 的**特征函数**。

定义 1.2.4 设 n 为任意固定的正整数，P 是 n 元谓词，函数

$$C_P(x_1, \cdots, x_n) = \begin{cases} 1 & \text{当 } P(x_1, \cdots, x_n) \text{为真}, \\ 0 & \text{当 } P(x_1, \cdots, x_n) \text{为假}。\end{cases}$$

称为 n 元谓词 P 的**特征函数**。

由定义容易看出，特征函数都是全函数；n 元谓词的特征函数就是该谓词的外延集的特征函数。

定义 1.2.5 设 f 为 n 元部分函数，函数

$$g(x_1,\cdots,x_n)=\begin{cases}f(x_1,\cdots,x_n) & \text{如果}(x_1,\cdots,x_n)\in \mathrm{dom}\ f,\\ 0 & \text{如果}(x_1,\cdots,x_n)\notin \mathrm{dom}\ f。\end{cases}$$

称为 f 的**补全函数**。

补全函数当然都是全函数。全函数的补全函数就是它自己。今后，当 $(x_1,\cdots,x_n)\in \mathrm{dom}\ f$ 时，我们就说 $f(x_1,\cdots,x_n)$ 有定义，记作 $f(x_1,\cdots,x_n)\downarrow$；当 $(x_1,\cdots,x_n)\notin \mathrm{dom}\ f$ 时，我们就说 $f(x_1,\cdots,x_n)$ 无定义，记作 $f(x_1,\cdots,x_n)\uparrow$。

对于两个 n 元函数 f 和 g，当我们说 $f=g$ 时，我们的意思是下面两条同时成立：

1. $\mathrm{dom}\ f=\mathrm{dom}\ g$；

2. 当 $f(x_1,\cdots,x_n)$ 和 $g(x_1,\cdots,x_n)$ 都有定义时，函数值相等，即

$$f(x_1,\cdots,x_n)=g(x_1,\cdots,x_n)。$$

第二章

图灵可计算函数

本世纪三十年代，A·M·图灵(Turing)研究了人作笔算时的动作，他把这些动作分成两类：一是在纸上写下或抹掉一个符号，二是将目光从纸上的一处转移到另一处。根据这种分解，他提出了一种理论上的计算模型，称为图灵机。迄今为止，在为数众多的理论计算模型中，图灵机是使用最广、影响最大的一种。图灵的第一篇论文发表于1936年，此后波斯特等人又对图灵的理论作了若干改进。

图灵机可以看作是一种抽象的符号演算，也可以设想为一种物化的机械装置。后一种处理在直观上比较清楚，我们就从这种直观入手来进行讨论。

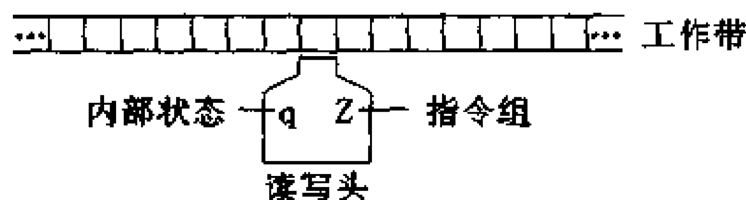


图 2.1　图灵机

图灵机的"机械部分"包括两项内容：

1. 一条可以向两端无限伸长的带子，称为**工作带**。工作带上划成大小相同的方格，每个方格中可以容纳一个字母。

2. 一个可以沿工作带左右移动(每次移动一格)的**读写头**。读写头在不同的时刻可以处于不同的内部状态。读写头有如下三

项功能：① 认出它所对的方格中的字母，② 在它所对的方格中写下一个字母(同时清除该方格中原有的字母)，③ 左移一格或右移一格。

除了“机械部分”之外，每部图灵机还有一组指令或叫程序，它们负责指挥读写头的动作。实际上，这部图灵机与那部图灵机的区别仅在于它们的指令组不同，定义一部具体的图灵机也就是具体给出它的指令组(程序)。因此我们今后就将图灵机与其指令组等同看待。

§2.1 图灵机的基本概念

为了严格陈述与图灵机有关的定义，我们需要建立一些基本概念。

1. 字母表 一个可数无穷的初始符号集

$\{S_0, S_1, S_2, \cdots\}$

称为字母表，其元素称为**字母**。直观地讲，S_0 表示工作带上的空格，也简记作 0；S_1 也简记作 1。一部图灵机只会用到有穷多个字母，但至少要有两个字母才能进行有意义的计算。

2. 内部状态 一个可数无穷的初始符号集

$\{q_0, q_1, q_2, \cdots\}$

称为内部状态集，其元素称为**内部状态**，简称**状态**。一部图灵机只会用到有穷多个状态。状态的直观意义是：读写头在不同状态下读到同一个字母时可以采取不同的动作。

3. (初始符号)R 和 L 直观上讲，R 表示读写头向右移一格，L 表示读写头向左移一格。

4. 符号串 由以上三种初始符号所组成的有穷序列称为**符号串**。

定义 2.1.1 具有以下三种形状的符号串称为**四元组**，也叫**指令**：

$q_i\, S_j\, S_k\, q_l$

$q_i\, S_j\, R\, q_l$

$q_i\, S_j\, L\, q_l$

其中 S_j,S_k 是任何字母，q_i,q_l 是任何内部状态。

三种四元组的直观意义是：

$q_i\, S_j\, S_k\, q_l$　表示当读写头处于状态 q_i、在工作带上读到 S_j 时，执行以下动作：将该方格中的 S_j 改为 S_k，读写头位置不动，其状态变为 q_l。

$q_i\, S_j\, R\, q_l$　表示当读写头处于状态 q_i、在工作带上读到 S_j 时，执行以下动作：读写头向右移一格，其状态变为 q_l(工作带上内容不变)。

$q_i\, S_j\, L\, q_l$　表示当读写头处于状态 q_i、在工作带上读到 S_j 时，执行以下动作：读写头向左移一格，其状态变为 q_l(工作带上内容不变)。

定义 2.1.2　仅由字母组成的非空的符号串成为**带表达式**。

从直观上讲，当我们说图灵机在某一时刻的带表达式为

$$S_{i_1}\, S_{i_2}\, S_{i_3} \cdots\, S_{i_m}$$

时，就是说当时工作带上有一段长度为 m 的部分，其 m 个方格中依次写有字母 S_{i_1}，S_{i_2}，S_{i_3}，…，S_{i_m}，而其余方格都是空格，即视为有字母 S_0(也就是 0)。

定义 2.1.3　由若干(至少一个)字母和恰好一个内部状态 q_i 所组成的符号串，如果 q_i 不是最右边的符号，就称之为一个**瞬时描述**。

从直观上讲，当我们说某个图灵机在某一时刻的瞬时描述是

$$S_{j_1} \cdots S_{j_m}\, q_i\, S_{j_{m-1}} \cdots S_{j_n}$$

时，意味着此时该图灵机的带表达式为

$$S_{j_1} \cdots S_{j_m}\, S_{j_{m+1}} \cdots S_{j_n}$$

而读写头正对着写有 $S_{j_{m+1}}$ 的那个方格，状态为 q_i。以下，为了更加

形象、直观，我们将上述瞬时描述表示为

$$S_{j_1}\cdots S_{j_m}\underset{\substack{\uparrow\\ q_i}}{S_{j_{m+1}}}\cdots S_{j_n}$$

为了用图灵机计算 n 元函数 $f(x_1, x_2, \cdots x_n)$，我们必须规定在计算开始时如何向图灵机输入 n 元数组$(x_1, x_2, \cdots x_n)$。

定义 2.1.4　对任意固定的正整数 n，用下述带表达式

$$\underbrace{11\cdots1}_{x_1+1\text{个}}0\underbrace{11\cdots1}_{x_2+1\text{个}}0\cdots\cdots0\underbrace{11\cdots1}_{x_n+1\text{个}}$$

表示输入的 n 元数组$(x_1, x_2, \cdots x_n)$。这个带表达式也记作

$$\overline{(x_1, x_2, \cdots, x_n)}$$

或

$$1^{(x_1+1)}01^{(x_2+1)}0\cdots\cdots01^{(x_n+1)}。$$

其中的记号 $1^{(x_i+1)}$表示连续 x_i+1 个字母 1(即 S_1)。

之所以用 $1^{(x+1)}$表示 x 而不用 $1^{(x)}$表示 x，是为了使输入 0 时的带表达式有别于空带。

定义 2.1.5　形如

$$q_0\,\overline{(x_1, \cdots, x_n)}$$

的瞬时描述称为**初始的瞬时描述**。

直观地说，初始的瞬时描述表示对图灵机的输入：输入 n 元数组$(x_1, \cdots, x_n)$后，读写头的内部状态总为 q_0，读写头的位置总是对着输入的最左边的那个 1。

定义 2.1.6　一个由四元组组成的有穷集合，如果其中没有两个四元组的前两个符号是对应相同的，就称之为一部**图灵机**(或一个**指令组**，或一个**程序**)。

按照这个定义，空集合是图灵机；集合$\{q_0\ 0\ 0\ q_0\}$是图灵机；集合$\{q_0\ 0\ 1\ q_1, q_0\ 1\ R\ q_0\}$也是图灵机。但$\{q_0\ 1\ 0\ q_0, q_0\ 1\ R\ q_1\}$却不是图灵机。定义规定图灵机中不能有两个四元组的前两个符号对应相同，是为了保证运算的确定性，或者说是为了保证所得到的形式

系统是单演的。

定义 2.1.7　设 Z 是图灵机，Z 的(一步)**运算**是从瞬时描述到瞬时描述的(部分)映射，分为三种情形：

(1) 如果瞬时描述 α 形如

$$S_{a_1}\cdots S_{a_m}\, q_i\, S_j\, S_{b_1}\cdots S_{b_n},$$

Z 中有四元组

$$q_i\, S_j\, S_k\, q_l,$$

则将 α 映到如下形状的瞬时描述 β：

$$S_{a_1}\cdots S_{a_m}\, q_l\, S_k\, S_{b_1}\cdots S_{b_n};$$

(2) 如果瞬时描述 α 形如

$$S_{a_1}\cdots S_{a_m}\, q_i\, S_j\, S_{b_1}\cdots S_{b_n},$$

Z 中有四元组

$$q_i\, S_j\, R\, q_l,$$

则将 α 映到如下形状的瞬时描述 β：

$$S_{a_1}\cdots S_{a_m}\, S_j\, q_l\, S_{b_1}\cdots S_{b_n};$$

(3) 如果瞬时描述 α 形如

$$S_{a_1}\cdots S_{a_{m-1}}\, S_{a_m}\, q_i\, S_j\, S_{b_1}\cdots S_{b_n},$$

Z 中有四元组

$$q_i\, S_j\, L\, q_l,$$

则将 α 映到如下形状的瞬时描述 β：

$$S_{a_1}\cdots S_{a_{m-1}}\, q_l\, S_{a_m}\, S_j\, S_{b_1}\cdots S_{b_n}。$$

如果瞬时描述 α 经图灵机 Z 的一步运算得到 β，就记作

$$\alpha\to\beta\ (Z)。$$

直观上讲，图灵机 Z 的一步运算就是依据该图灵机的指令在工作带上进行一次操作。

定义 2.1.8　如果图灵机 Z 中没有以 $q_i\, S_j$ 开头的指令，则形如

$$\cdots\cdots q_i\, S_j\cdots\cdots$$

的瞬时描述称为(对于 Z)**终止的瞬时描述**。

从直观上讲,如果图灵机 Z 运算到某一步时出现了终止的瞬时描述,运算就不能再继续下去了(不再有合适的指令可用),称为**停机**。

定义 2.1.9 图灵机 Z 的(一个)**计算**指的是由瞬时描述所组成的非空的有穷序列

$$\alpha_1,\ \alpha_2,\ \cdots,\ \alpha_m \qquad (m\geqslant 1)$$

满足

(1) α_1 是初始的瞬时描述;

(2) 当 $m>1$ 时,对每个 $i(1\leqslant i<m)$

$\alpha_i \rightarrow \alpha_{i+1}(Z)$;

(3) α_m 是终止的瞬时描述。

这里需要强调一下:定义中的(3)是不可少的,只有最终停止(停机)的一系列运算才构成一个计算。

例如,当 $Z=\{q_0\,1\,R\,q_0,\ q_0\,0\,R\,q_1,\ q_1\,1\,0\,q_2,\ q_1\,0\,R\,q_1\}$时,如果输入二元数组(1,2),就可以得到一个计算:

α_1　　110111
↑
q_0

α_2　　1 10111　　(由 α_1 根据 $q_0\,1\,R\,q_0$)
↑
q_0

α_3　　11 0111　　(由 α_2 根据 $q_0\,1\,R\,q_0$)
↑
q_0

α_4　　110 111　　(由 α_3 根据 $q_0\,0\,R\,q_1$)
↑
q_1

α_5　　110 011　　(由 α_4 根据 $q_1\,1\,0\,q_2$)
↑
q_2

至此，由于 Z 中没有以 $q_2 0$ 开头的指令，运算结束，形成了一个计算。

但是，如果向 Z 输入单个的自然数 1，就不能得到一个计算：读写头将一味右移，永不停止。

定义 2.1.10　设 β 是符号串，以 $|\beta|$ 表示 β 中 1（即字母 S_1）的个数。如果 $\alpha_1, \alpha_2, \cdots, \alpha_m$ 是图灵机 Z 的一个计算，则称 $|\alpha_m|$ 是该计算的**输出**。

§2.2　图灵可计算函数

给定图灵机 Z，当我们向 Z 输入 n 元数组 $(x_1, \cdots, x_n)$ 之后，在 Z 的指令指挥下，从初始的瞬时描述

$$q_0 1^{(x_1+1)} 0 \cdots\cdots 0 1^{(x_n+1)}$$

开始进行运算，有两种可能：

1. 经若干步运算后得到一个终止的瞬时描述 α_m，形成一个计算，有输出 $|\alpha_m|$；

2. 运算一直进行，永不停止，没有输出。

这样，图灵机的输入与输出之间就形成了一个部分函数关系，这个函数就叫该图灵机所计算的 n 元函数。严格的定义为：

定义 2.2.1　设 Z 是图灵机，n 是任意固定的正整数，定义部分函数 $\Phi_Z^{(n)}$ 为：对任何 n 元数组 $(x_1, \cdots, x_n)$

(1) 如果存在 Z 的一个计算 $\alpha_1, \alpha_2, \cdots, \alpha_m (m \geqslant 1)$ 使

$$\alpha_1 = q_0 \overline{(x_1, \cdots, x_n)},$$

则令 $\Phi_Z^{(n)}(x_1, \cdots, x_n) = |\alpha_m|$；

(2) 如果不存在满足上述条件的计算，则规定 $\Phi_Z^{(n)}(x_1, \cdots, x_n)$ 无定义。

$\Phi_Z^{(n)}$ 称为**图灵机 Z 所计算的 n 元函数**。当 $n=1$ 时，上标可以略去不写，即将 $\Phi_Z^{(1)}$ 简记作 Φ_Z。

对任何 n 元部分函数（不管它是用什么方法定义的）f，如果

存在图灵机 Z 使得

$$f=\Phi_Z^{(n)},$$

则称 f 为**图灵可计算函数简称可计算函数**。

这里有两个细节需要强调一下：

1. 一部图灵机并不只计算一个函数，而是对于每个正整数 n 计算一个 n 元函数。Φ_Z，$\Phi_Z^{(2)}$，…，$\Phi_Z^{(n)}$，…都是 Z 所计算的函数。

2. 对每个图灵可计算函数 f，都有无穷多个图灵机可以计算它。例如上一节末的例子中的图灵机

$$Z=\{q_0\,1\,R\,q_0,\ q_0\,0\,R\,q_1,\ q_1\,1\,0\,q_2,\ q_1\,0\,R\,q_1\},$$

它所计算的二元函数是

$$f(x,\ y)=x+y+1。$$

如果向 Z 中加入一个新的四元组 $q_8\,0\,R\,q_1$，就得到了另一个图灵机 Z'，显然 Z' 所计算的二元函数仍是 $x+y+1$。因为新增的这个四元组根本用不上。

此外，根据上一节末的例子以及定义 2.2.1，我们已经看到二元函数 $x+y+1$ 是图灵可计算函数，处处无定义的一元函数也是图灵可计算函数(它们都是由 Z 计算的)。下面我们再来讨论一些简单的图灵可计算函数。

例 1 考虑图灵机

$$Z_S=\{q_0\,1\,L\,q_0\}。$$

当向 Z_S 输入单个自然数 x 之后，初始的瞬时描述为

$$\alpha_1 \qquad 0\,\underset{\uparrow\atop q_0}{1}\overbrace{1\cdots1}^{x+1\text{个}}$$

在 Z_S 中唯一一条指令的指挥下进行运算，得到

$$\alpha_2 \qquad \underset{\uparrow\atop q_0}{0}\,\overbrace{11\cdots1}^{x+1\text{个}}$$

由于 Z_S 中没有以 $q_0\,0$ 开头的指令，所以 α_2 是终止的瞬时描述，输出为 $x+1$。就是说

$$\Phi_{Z_S}(x)=x+1。$$

从而后继函数

$$S(x)=x+1$$

是图灵可计算函数(此外，空图灵机也是计算后继函数的)。

例 2 考虑图灵机

$$Z_O=\{q_0\,1\,0\,q_1,\ q_1\,0\,R\,q_0\}。$$

当输入单个自然数 x 后，Z_O 的计算过程为

α_1	$q_0\,1^{(x+1)}$
α_2	$q_1\,0\,1^{(x)}$
α_3	$0\,q_0\,1^{(x)}$
α_4	$0\,q_1\,0\,1^{(x-1)}$
	…………
α_{2x+4}	$0^{(x+1)}q_0\,0$ (停机)。

Z_O 的作用就是将输入的 1 全部抹光，因而最后的输出总是 0，于是

$$\Phi_{Z_O}(x)=O(x)=0$$

从而(一元的)**零函数**

$$O(x)=0$$

是图灵可计算函数。

例 3 二元函数 $x+y$ 是图灵可计算函数。

由于输入二元数组$(x,\ y)$的带表达式为：

$$\overbrace{11\cdots1}^{x+1\text{个}}\,0\,\overbrace{11\cdots1}^{y+1\text{个}}$$

总共 $x+y+2$ 个 1，所以计算二元函数 $x+y$ 的图灵机 Z_+ 的任务就是在前后两个部分的 1 中各抹掉一个 1，然后停机。令

$$Z_+=\{q_0\,1\,0\,q_0,\ q_0\,0\,R\,q_1,\ q_1\,1\,R\,q_1,\ q_1\,0\,R\,q_2,\ q_2\,1\,0\,q_2\}$$

容易验证

$\Phi^{(2)}_{Z_+}(x,y)=x+y$。

例 4 二元函数 $x-y$ 是图灵可计算函数。

由于 $x-y$ 是定义域不全的函数(仅当 $x\geqslant y$ 时有定义),因此计算 $x-y$ 的图灵机要满足以下两个条件:

1. 当 $x\geqslant y$ 时,输入二元数组(x, y)之后能够停机,其终止的带表达式上恰好有 $x-y$ 个 1;

2. 当 $x<y$ 时永不停机。

我们按以下的思路来构造 Z_-:

输入(x, y)之后,每一轮次在$\overline{(x, y)}$的 x 部分和 y 部分对等地一边抹掉一个 1。如果 y 部分先抹光或者两部分在同一轮次抹光,停机;如果 x 部分先抹光,读写头不停地向左移动。

Z_-可以由以下四元组构成:

$q_0\ 1\ 0\ q_1$ (抹掉 x 部分最左边的 1)

$$\left.\begin{array}{l} q_1\ 0\ R\ q_2 \\ q_2\ 1\ R\ q_2 \\ q_2\ 0\ R\ q_3 \\ q_3\ 1\ R\ q_3 \\ q_3\ 0\ L\ q_4 \end{array}\right\}\ \text{(找到 } y \text{ 部分最右边的 1)}$$

$q_4\ 1\ 0\ q_4$ (抹掉这个 1)

$$\left.\begin{array}{l} q_4\ 0\ L\ q_5 \\ q_5\ 1\ L\ q_6 \end{array}\right\}\ \text{(如果 } y \text{ 部分还有 1,左移;如果 } y \text{ 部分已经没有 1 了,停机)}$$

$$\left.\begin{array}{l} q_6\ 1\ L\ q_6 \\ q_6\ 0\ L\ q_7 \\ q_7\ 1\ L\ q_7 \\ q_7\ 0\ R\ q_0 \end{array}\right\}\ \text{(找 } x \text{ 部分最左边的 1,如果还有,重复前面的过程)}$$

$q_0\ 0\ L\ q_0$ (如果 x 部分的 1 都已抹光,左移,永不停止。)

例 5 二元函数 $2(x+1)$是图灵可计算函数。

对于函数 $f(x)=2(x+1)$，输入 x 后的带表达式为一串 $x+1$ 个 1，而输出的带表达式应当是 $2(x+1)$ 个 1。因此计算函数 $2(x+1)$ 的图灵机 Z_2 的作用就是将工作带上的 1 逐个复制一遍。

Z_2 可以由以下四元组构成：

$q_0\ 1\ R\ q_0$
$q_0\ 0\ R\ q_1$
$q_1\ 0\ 1\ q_1$
（①找到 $1^{(x+1)}$ 右边的第二个 0，将它改为 1）

$q_1\ 1\ L\ q_2$
$q_2\ 1\ L\ q_2$
$q_2\ 0\ L\ q_3$
$q_3\ 0\ L\ q_3$
$q_3\ 1\ L\ q_4$
$q_4\ 1\ R\ q_5$
（②左移，看输入部分是否还有不只一个 1，如有，找到最右边的那个 1）

$q_5\ 1\ 0\ q_5$ （将这个 1 抹掉）

$q_5\ 0\ R\ q_6$
$q_6\ 0\ R\ q_6$
$q_6\ 1\ R\ q_7$
$q_7\ 1\ R\ q_7$
$q_7\ 0\ 1\ q_2$
（③找到带上最右边的 1 右边的 0，将它改为 1 然后回到②）

$q_4\ 0\ R\ q_8$
$q_8\ 1\ 0\ q_8$
$q_8\ 0\ R\ q_9$
$q_9\ 0\ 1\ q_{10}$
$q_{10}\ 1\ R\ q_9$
（④如果输入部分只剩下一个 1，将它抹掉，然后将其右边的 $x+1$ 个 0 都改成 1，停机。）

在以上各例中，我们只用到 0，1 两个字母。实际上，为了计算函数，只要这两个字母也就够了。不过这对我们并不重要，在许多时候我们宁愿多用几个字母，这可以使图灵机变得简单一些——可以减少内部状态的个数和指令的个数。例如下述含有四个字母

0, 1, ϵ, δ 的图灵机 Z'_2 也是计算函数 $2(x+1)$的(请读者自己验证一下),它由以下指令构成:

$q_0\ 1\ \epsilon\ q_0$ (将第一个 1 改为 ϵ)

$$\left.\begin{array}{l} q_0\ \epsilon\ R\ q_1 \\ q_1\ 1\ R\ q_1 \\ q_1\ \delta\ R\ q_1 \\ q_1\ 0\ \delta\ q_2 \end{array}\right\}$$ (右移,找到第一个 0,将它改为 δ)

$$\left.\begin{array}{l} q_2\ \delta\ L\ q_2 \\ q_2\ 1\ L\ q_2 \\ q_2\ \epsilon\ R\ q_0 \end{array}\right\}$$ (左移,找到 ϵ 右边的符号,如果是 1,重复以上过程)

$$\left.\begin{array}{l} q_0\ \delta\ 1\ q_3 \\ q_3\ 1\ R\ q_0 \end{array}\right\}$$ (如果 ϵ 右边是 δ,将所有 δ 都改成 1)

$$\left.\begin{array}{l} q_0\ 0\ L\ q_4 \\ q_4\ 1\ L\ q_4 \\ q_4\ \epsilon\ 1\ q_4 \end{array}\right\}$$ (左移,将所有的 ϵ 都改成 1,停机。)

Z'_2 中只有四个状态,比 Z_2 少得多,Z'_2 中的四元组也比 Z_2 少。在下一节中我们经常要用到 0,1 以外的字母。

§2.3 正 则 的 图 灵 机

上一节我们建立了一些函数的图灵可计算性。我们已经看到,即使是十分简单的函数,计算它们的图灵机也可能相当复杂。在递归论中,我们所关心的主要是一个函数**是否**图灵可计算函数,而并不怎么关心究竟如何去计算它们。因此我们在这一节要讨论图灵可计算函数的一些性质,以便能够使用各种间接的(但不那么烦琐的)方法去证明各种具体函数是(或者不是)图灵可计算函数。

设 Z 为图灵机,我们用 $\theta(Z)$表示 Z 的四元组中所出现的内部状态的最大下标;用 $Z^{(n)}$表示将 Z 中每个状态 q_i 都换成 q_{i+n}所得

到的图灵机。

例如，对例 2 中计算一元零函数的图灵机

$$Z_O=\{q_0\ 1\ 0\ q_1,\ q_1\ 0\ R\ q_0\}$$

$\theta(Z_O)=1$， $Z_O^{(n)}=\{q_n\ 1\ 0\ q_{n+1},\ q_{n+1}\ 0\ R\ q_n\}$。

定义 2.3.1 设 Z 为图灵机，n 为任意的正整数，如果

(1) 存在 $s>0$，使得只要对 Z 输入 n 元数组 $(x_1,\cdots,x_n)$ 时有输出，其终止的瞬时描述必定形如

$$\underbrace{11\cdots1}_{r_1\uparrow}0\underbrace{11\cdots1}_{r_2\uparrow}0\cdots\cdots0\underbrace{11\cdots1}_{r_s\uparrow}$$

$$\uparrow$$

$$q_{\theta(Z)}$$

(其中 $r_1,\ r_2,\ \cdots,\ r_s\geqslant0$)；

(2) Z 中没有以 $q_{\theta(Z)}$ 开头的四元组。

则称 Z 为 n **正则图灵机**。在不致引起混淆的情形，n 可以省略，简称**正则图灵机**。

正则图灵机的最显著的特点，是其终止的瞬时描述具有初始瞬时描述的形状。

在一般情形下，对于两部图灵机 Z_1 和 Z_2，将它们的指令合在一起所得到的指令集 $Z_1\cup Z_2$ 未必仍是图灵机，因为 $Z_1\cup Z_2$ 中很可能会有两个四元组的头两个符号相同；如果我们将 Z_2 换成 $Z_2^{(m)}$ (其中 $m=\theta(Z_1)$)，所得到的并集即 $Z_1\cup Z_2^{(m)}$ 仍然不一定是图灵机，因为 Z_1 中可能有以 q_m 开头的四元组。但是，如果 Z_1 是正则图灵机，$Z_1\cup Z_2^{(m)}$ 就一定是图灵机，而且在输入 n 元数组后一定是先对这个 n 元数组运行 Z_1，然后再对 Z_1 所输出的 s 元数组运行 Z_2 (如果 Z_1 能停机的话)；如果 Z_1，Z_2 都是正则图灵机，那么 $Z_1\cup Z_2^{(m)}$ 就也是正则图灵机。

今后对于正则图灵机 Z_1 和 Z_2，我们将 $Z_1\cup Z_2^{(m)}$ (其中 $m=\theta(Z_1)$) 记作

$$Z_1*Z_2。$$

定理 2.3.1　任给图灵机 Z，存在图灵机 Z' 使得

(1) 对每个正整数 n，Z' 都是 n 正则的；

(2) 对每个正整数 n，

$$\Phi_Z^{(n)}=\Phi_{Z'}^{(n)}。$$

证　任给图灵机 Z，我们按以下要求来构造 Z'：

1. 分别使用 Z 中未用到的两个字母（记作 ξ 和 η），在输入的数据 $(x_1,\cdots,x_n)$ 的两端各设一个标志，使带表达式变为

$$\xi\,\underbrace{11\cdots1}_{x_1+1个}\,0\,\underbrace{11\cdots1}_{x_2+1个}\,0\cdots\cdots0\,\underbrace{11\cdots1}_{x_n+1个}\,\eta;$$

2. 改造 Z，使 Z 的全部运算都在 ξ 和 η 之间进行。具体做法是：如果读写头移到写有 ξ 的方格，就将 ξ 向左移一格，如果读写头移到写有 η 的方格，则将 η 向右移一格，然后再回到 Z 的运算；

3. Z 停机后，如果 ξ 和 η 之间有 1，将这些 1 都移到一起（即令定义 2.3.1 中的 $s=1$），然后抹掉 ξ 和 η，再将读写头移到最左边的 1 上，进入一个新状态，停机；如果没有 1，抹掉 ξ 和 η 后进入一个新状态，停机；

4. 如果 Z 不能停机，Z' 也不能停机。

Z' 可以由以下几个部分的四元组构成（其中 ξ 和 η 是 Z 中未用到的两个字母）：

$$\left.\begin{array}{l} q_0\,1\,L\,q_0 \\ q_0\,0\,\xi\,q_0 \end{array}\right\}\quad（在左端设标志\ \xi）$$

$$\left.\begin{array}{l} q_0\,\xi\,R\,q_1 \\ q_1\,1\,R\,q_1 \\ q_1\,0\,R\,q_2 \\ q_2\,1\,R\,q_1 \\ q_2\,0\,L\,q_3 \end{array}\right\}\quad\begin{array}{l}（找到输入数据的右端，即找到\\ \quad 连续两个\ 0\ 中左边的一个）\end{array}$$

$$q_3\,0\,\eta\,q_3\qquad（将这个\ 0\ 改为\ \eta）$$

$$\left.\begin{array}{l} q_3\ \eta\ L\ q_4 \\ q_4\ 1\ L\ q_4 \\ q_4\ 0\ L\ q_4 \\ q_4\ \xi\ R\ q_5 \end{array}\right\}$$ （读写头回到最左边的 1 上）

以上指令构成的 Z_1 实现了要求 1。

在 $Z^{(5)}(\theta(Z_1)=5)$ 中增加一些指令，得到 Z_2（其中 $m=\theta(Z^{(5)})$）：对 $Z^{(5)}$中出现的每个 q_i，增加

$$\left.\begin{array}{l} q_i\ \xi\ 0\ q_{i+m} \\ q_{i+m}0\ L\ q_{i+2m} \\ q_{i+2m}0\ \xi\ q_{i+2m} \\ q_{i+2m}\xi\ R\ q_i \end{array}\right\}$$ （将 ξ 向左移一格，然后回到 $Z^{(5)}$的运算）

$$\left.\begin{array}{l} q_i\ \eta\ 0\ q_{i+3m} \\ q_{i+3m}0\ R\ q_{i+4m} \\ q_{i+4m}0\ \eta\ q_{i+4m} \\ q_{i+4m}\eta\ L\ q_i \end{array}\right\}$$ （将 η 向右移一格，然后回到 $Z^{(5)}$的运算）

Z_2 满足了要求 2 和要求 4。

令 $M=\theta(Z_2)+1$，我们来构造 Z_3 以满足要求 3。

Z_3：

对每个 $i<M$，如果 Z_2 中有 q_i 和 S_j，但没有以 $q_i S_j$ 开头的四元组（即 Z_2 可能在状态 q_i 下读到 S_j 时停机），则 Z_3 中有指令

$$q_i\ S_j\ S_j\ q_M$$

此外，对于 Z 中所出现的、不等于 0,1 的任何字母 S，Z_3 中有如下指令：

$$\left.\begin{array}{l} q_M\ 1\ L\ q_M \\ q_M\ 0\ L\ q_M \\ q_M\ S\ L\ q_M \\ q_M\ \xi\ R\ q_{M+1} \end{array}\right\}$$ （找到 ξ）

$$\left.\begin{array}{l} q_{M+1}0\ R\ q_{M+1} \\ q_{M+1}S\ 0\ q_{M+1} \end{array}\right\}$$ （①右移，抹掉 0,1 以外的字母，找到 1 或 η）

$q_{M+1}1\ 0\ q_{M+2}$ （如果找到 1，抹掉）

$\left.\begin{array}{l} q_{M+2}0\ L\ q_{M+2} \\ q_{M+2}1\ R\ q_{M+3} \\ q_{M+2}\xi\ R\ q_{M+3} \end{array}\right\}$ （②左移，找到 ξ 或紧挨着 ξ 的一串 1 中的最右面的那个 1）

$\left.\begin{array}{l} q_{M+3}0\ 1\ q_{M+3} \\ q_{M+3}1\ R\ q_{M+1} \end{array}\right\}$ （将找到的 ξ 或 1 右边的 0 改为 1，回到①）

$q_{M+1}\eta\ 0\ q_{M+4}$ （如果①找到的是 η，把它抹掉）

$\left.\begin{array}{l} q_{M+4}0\ L\ q_{M+4} \\ q_{M+4}1\ L\ q_{M+4} \\ q_{M+4}\xi\ 0\ q_{M+5} \end{array}\right\}$ （左移，找到 ξ，抹掉）

$q_{M+5}0\ R\ q_{M+6}$ （右移一格，进入新状态，停机）

不难验证

$$Z'=Z_1\cup Z_2\cup Z_3$$

就是定理所要求的正则图灵机。

定理 2.3.1 使得我们可以只限于讨论正则图灵机。

定理 2.3.2　对每个 n 正则图灵机 Z，存在图灵机 Z_p，使得

(1) 对任何 $p>0$，Z_p 都是 $p+n$ 正则的；

(2) 如果$\Phi_Z^{(n)}(x_1,\cdots,x_n)$有定义，其终止的瞬时描述为

$$q_{\theta(Z)}1^{(r_1)}0\cdots01^{(r_s)},$$

则对任意的 $y_1,\cdots,y_p$，$\Phi_{Z_p}^{(p+n)}(y_1,\cdots,y_p,x_1,\cdots,x_n)$也有定义，且其终止的瞬时描述为

$$q_{\theta(Z_p)}1^{(y_1+1)}0\cdots01^{(y_p+1)}01^{(r_1)}0\cdots01^{(r_s)}$$

(3) 如果 $\Phi_Z^{(n)}(x_1,\cdots,x_n)$ 无定义，则对任何 $y_1,\cdots,y_p$，$\Phi_{Z_p}^{(p+n)}(y_1,\cdots,y_p,x_1,\cdots,x_n)$也无定义。

证明定理 2.3.2 的办法是在 Z 的基础上构造图灵机 Z_p。其主要思路是

1. 先在输入的 y 部分的两端设置标志 ξ 和 η（ξ 和 η 是不在 Z 中出现的两个字母），使带表达式变为

$$\xi 1^{(y_1+1)}0\cdots 01^{(y_p+1)}\eta 1^{(x_1+1)}0\cdots 01^{(x_n+1)};$$

2. 转入 Z 的运算，并使每当读写头读到 η 时，将 ξ 和 η 之间的内容依次左移一格；

3. 当 Z 的运算停止时，如果读写头所对的方格中是 1，左移，找到 η，然后将 ξ，η 及其之间的内容依次右移，直到 η 右边是 1 为止。然后抹掉 ξ 和 η，读写头进入新状态，停在最左边的 1 上。

4. 当 Z 的运算停止时，如果读写头对着 0（即 $\Phi_Z^{(n)}(x_1,\cdots,x_n)=0$），抹掉 ξ 和 η，读写头进入新状态，停在最右边的 1 上。

具体的构造留给读者（注意：n 是固定的，p 是任意的！参看定理 2.3.1）

从直观上讲，Z_p 对右边 n 个分量的作用与 Z 完全一样，但保持左边的 p（p 是任意的！）个分量不动。

以下，我们来讨论几种具体的图灵机，它们将在下一节中充当重要角色。

例 2.3.1　右 n 复制机 CR_n

CR_n 是正则图灵机，其作用是将输入的 $n+p$ 元数组的带表达式

$$\overline{(x_1,\cdots,x_n,y_1,\cdots,y_p)}$$

变成

$$\overline{(x_1,\cdots,x_n,y_1,\cdots,y_p,x_1,\cdots,x_n)}$$

然后停机。即将输入的左边 n 个分量在右边复制一遍。

CR_n 可以由三个部分构成：

1. 在带表达式

$$\overline{(x_1,\cdots,x_n,y_1,\cdots,y_p)}$$

的左端设置一个标志 ξ，右端设置一个标志 η，在 x_n 和 y_1 之间设置一个标志 ω（ξ，η 和 ω 都是在 Z 中不出现的字母），使带表达式变为

$$\xi 1^{(x_1+1)}0\cdots 01^{(x_n+1)}\omega 1^{(y_1+1)}0\cdots 01^{(y_p+1)}\eta$$

2. 在 η 右边复制 ξ 和 ω 之间的内容。具体做法是：找到 ξ 与 ω

之间的 1，改成 ε，然后在 η 右边的相应位置上写下一个 1；找到 ξ 与 ω 之间的 0，改成 λ，然后在 η 右边相应的位置上写下一个 δ。实际过程是：

读写头向左移，找到右起第一个 ε，λ 或 ξ，根据它的右边是 1，0 还是 ω 而分别采取以下步骤：

(1) 如果是 1，将它改为 ε，右移，找到连续两个 0，将左边的那个 0 改为 1，返回；

(2) 如果是 0，将它改为 λ，右移，找到连续两个 0，将左边的那个 0 改为 δ，返回；

(3) 如果是 ω，右移，找到连续两个 0，再左移，一路上将所遇到的 $\lambda,\delta,\eta,\omega$ 全部抹掉；将所有的 ε 都改为 1；最后抹掉 ξ，右移一格，进入新状态，停机。

CR_n 由以下指令构成：

$$\left.\begin{array}{l} q_0\,1\,L\,q_0 \\ q_0\,0\,\xi\,q_0 \end{array}\right\} \quad \text{(置 } \xi\text{)}$$

$$\left.\begin{array}{l} q_0\,\xi\,R\,q_1 \\ q_1\,1\,R\,q_1 \\ q_1\,0\,R\,q_2 \\ q_2\,1\,R\,q_1 \\ q_2\,0\,L\,q_3 \\ q_3\,0\,\eta\,q_4 \end{array}\right\} \quad \text{(找到右端，即找到连续两个 0 中左边的那一个，改为 } \eta\text{)}$$

$$\left.\begin{array}{l} q_4\,\eta\,L\,q_4 \\ q_4\,1\,L\,q_4 \\ q_4\,0\,L\,q_4 \\ q_4\,\xi\,R\,q_5 \end{array}\right\} \quad \text{(左移，找到左起第一个 1)}$$

$q_5\ 1\ R\ q_5$
$q_5\ 0\ R\ q_6$
$q_6\ 1\ R\ q_6$
$q_6\ 0\ R\ q_7$
……
$q_{n+4} 1\ R\ q_{n+4}$
$q_{n+4} 0\ \omega\ q_{n+5}$
$q_{n+4} \eta\ \omega\ q_{n+5}$

（找到 x 部分和 y 部分之间的那个 0，将它改为 ω。如果 $p=0$，则 ω,η 的位置重合，保留 ω）

$q_{n+5} \omega\ L\ q_{n+5}$
$q_{n+5} 1\ L\ q_{n+5}$
$q_{n+5} 0\ L\ q_{n+5}$
$q_{n+5} \delta\ L\ q_{n+5}$
$q_{n+5} \eta\ L\ q_{n+5}$
$q_{n+5} \varepsilon\ R\ q_{n+6}$
$q_{n+5} \lambda\ R\ q_{n+6}$
$q_{n+5} \xi\ R\ q_{n+6}$

（左移，找到最右边的 ϵ,λ 或 ξ，视其右方为 $1,0,\omega$ 而分别采取三种不同措施）

$q_{n+6} 1\ \epsilon\ q_{n+7}$
$q_{n+7} \epsilon\ R\ q_{n+7}$
$q_{n+7} 1\ R\ q_{n+7}$
$q_{n+7} \omega\ R\ q_{n+7}$
$q_{n+7} \eta\ R\ q_{n+7}$
$q_{n+7} \delta\ R\ q_{n+7}$
$q_{n+7} 0\ R\ q_{n+10}$
$q_{n+10} 1\ R\ q_{n+7}$
$q_{n+10} 0\ L\ q_{n+13}$
$q_{n+13} 0\ 1\ q_{n+5}$

（将 x 部分最右边的 1 改为 ϵ，移至右端，写下一个 1）

$$\left.\begin{array}{l}q_{n+6}0\ \lambda\ q_{n+8}\\ q_{n+8}\lambda\ R\ q_{n+8}\\ q_{n+8}1\ R\ q_{n+8}\\ q_{n+8}\omega\ R\ q_{n+8}\\ q_{n+8}\eta\ R\ q_{n+8}\\ q_{n+8}\delta\ R\ q_{n+8}\\ q_{n+8}0\ R\ q_{n+11}\\ q_{n+11}1\ R\ q_{n+8}\\ q_{n+11}0\ L\ q_{n+14}\\ q_{n+14}0\ \delta\ q_{n+5}\end{array}\right\}$$

（将 x 部分的 0 改为 λ，移至右端，写下一个 δ）

$$\left.\begin{array}{l}q_{n+6}\omega\ R\ q_{n+9}\\ q_{n+9}1\ R\ q_{n+9}\\ q_{n+9}\delta\ R\ q_{n+9}\\ q_{n+9}\eta\ R\ q_{n+9}\\ q_{n+9}0\ R\ q_{n+12}\\ q_{n+12}1\ R\ q_{n+9}\\ q_{n+12}0\ L\ q_{n+15}\end{array}\right\}$$

（复制结束，找到右端）

$$\left.\begin{array}{l}q_{n+15}0\ L\ q_{n+15}\\ q_{n+15}1\ L\ q_{n+15}\\ q_{n+15}\delta\ 0\ q_{n+15}\\ q_{n+15}\eta\ 0\ q_{n+15}\\ q_{n+15}\omega\ 0\ q_{n+15}\\ q_{n+15}\lambda\ 0\ q_{n+15}\\ q_{n+15}\varepsilon\ 1\ q_{n+15}\\ q_{n+15}\xi\ 0\ q_{n+16}\\ q_{n+16}0\ R\ q_{n+17}\end{array}\right\}$$

（左移，一路上将 $\delta,\eta,\lambda,\omega,\xi$ 都抹掉，将 ε 改为 1，进入新状态，停在最左边的 1 上）

与此类似，我们也有**左 n 复制机** CL_n，在向它输入 $p+n(p\geqslant 0)$ 元数组 $(y_1,\cdots,y_p,x_1,\cdots,x_n)$ 后，其终止的带表达式为

$$\overline{(x_1,\cdots,x_n,y_1,\cdots,y_p,x_1,\cdots,x_n)}。$$

例 2.3.2　**左 n 消去机 ML_n。**

ML_n 是正则图灵机，对任给的 $p\geq 0$，当输入 $n+p$ 元数组

$$(x_1,\cdots,x_n,y_1,\cdots,y_p)$$

后，其终止的带表达式为

$$\overline{(y_1,\cdots,y_p)}$$

也就是说，ML_n 的作用是将输入的前 n 个分量抹掉。具体构造留给读者。

类似地，我们也有**右 n 消去机** MR_n，其作用是抹掉输入的右边 n 个分量。

例 2.3.3　**左 n 反转机 IL_n**

IL_n 也是正则图灵机，在输入 $n+p$ 元数组

$$(x_1,\cdots,x_n,y_1,\cdots,y_p)$$

后，其终止的带表达式为

$$\overline{(y_1,\cdots,y_p,x_1,\cdots,x_n)}。$$

其作用是将左边的 n 个分量转移到右边。

不难看出：

$$IL_n=CR_n * ML_n$$

类似地，也有**右 n 反转机** IR_n。

例 2.3.4　**左 $n+1$ 增复制机 CL^+_{n+1}**

这也是一种正则图灵机，当输入 $n+1$ 元数组

$$(x_1,\cdots,x_n,y)$$

之后，其终止的带表达式为

$$\overline{(x_1,\cdots,x_n,y+1,x_1,\cdots,x_n,y)}$$

即在左边将输入的数据复制一遍，但第 $n+1$ 个分量要增加 1。

例 2.3.5　**右 $n+1$ 减复制机 C^-_{n+1}**

这也是一种正则图灵机，在输入 $n+1$ 元数组

$$(x_1,\cdots,x_n,y)$$

之后，终止的带表达式为

$\overline{(x_1,\cdots,x_n,y,x_1,\cdots,x_n,y-1,\cdots,x_1,\cdots,x_n,0,x_1,\cdots,x_n)}$

它的作用是将输入的数据在右边复制 $y+1$ 次，每次复制第 $n+1$ 个分量的带表达式都减 1。

例 2.3.4 和例 2.3.5 的构造都相当冗长，此处从略。相信读者能够在例 2.3.1 的启发下自己写出它们。

§2.4 可计算性与函数运算

在这一节，我们来讨论三种基本的函数运算——复合、原始递归和取极小（μ 运算）——以及图灵可计算性对这三种运算的封闭性。

复合是读者所熟悉的一种函数运算。不过由于在递归论中要处理定义域不全的函数，有必要再对复合函数的定义域做一点强调：

设 $g_1,\cdots,g_m$ 是 m 个 n 元（部分）函数，f 是 m 元（部分）函数，由它们复合而得的 n 元（部分）函数

$$h(x_1,\cdots,x_n)=f(g_1(x_1,\cdots,x_n),\cdots,g_m(x_1,\cdots,x_n))$$

有定义当且仅当

$$y_1=g_1(x_1,\cdots,x_n),$$
$$y_2=g_2(x_1,\cdots,x_n),$$
$$\cdots\cdots\cdots\cdots$$
$$y_m=g_m(x_1,\cdots,x_n)$$

和

$$z=f_1(y_1,\cdots,y_m)$$

都有定义。

对于复合函数，我们有

定理 2.4.1 设 $g_1,\cdots,g_m$ 是 m 个 n 元图灵可计算的（部分）函数，f 是 m 元图灵可计算的（部分）函数，由它们复合而得的 n 元函数

$$h(x_1,\cdots,x_n)=f(g_1(x_1,\cdots,x_n),\cdots,g_m(x_1,\cdots,x_n))$$

也是图灵可计算的。

证 由于 $g_1(x_1,\cdots,x_n)$，…，$g_m(x_1,\cdots,x_n)$都是图灵可计算的，所以 $g_1(x_1,\cdots,x_n)+1,\cdots,g_m(x_1,\cdots,x_n)+1$ 也都是图灵可计算的（见本章习题 3）。设 $Z_1,\cdots,Z_m$ 分别是计算$g_1(x_1,\cdots,x_n)+1,\cdots,g_m(x_1,\cdots,x_n)+1$ 的正则图灵机。Z_{ip}与 $Z_i(i=1,\cdots,m)$的关系如定理 2.3.2，M 为计算 $f(y_1,\cdots,y_m)$的正则图灵机，则

$$\underbrace{CR_n*\cdots*CR_n}_{m-1个}*Z_{mp}*IR_1*Z_{(m-1)p}*IR_1*\cdots*Z_{1p}*IR_1*M$$

是计算 $f(g_1(x_1,\cdots,x_n),\cdots,g_m(x_1,\cdots,x_n))$ 的图灵机。

其具体计算过程是：

第 1 步输入$(x_1,\cdots,x_n)$后，先将输入的带表达式复制 $m-1$ 次，使带表达式变成

$$\underbrace{\overline{(x_1,\cdots,x_n)}0\cdots0\overline{(x_1,\cdots,x_n)}}_{m组};$$

第 2 步用 Z_{mp}在最右边的一组$\overline{(x_1,\cdots,x_n)}$上计算 $g_m(x_1,\cdots,x_n)+1$，然后用 IR_1 将计算的结果（如果能停机的话）转移到带表达式的最左边；

第 3 步用 $Z_{(m-1)p}$在（此时的）最右边的一组$\overline{(x_1,\cdots,x_n)}$上计算 $g_{m-1}(x_1,\cdots,x_n)+1$，然后用 IR_1将计算的结果（如果能停机的话）转移到带表达式的最左边；

…………

第 m+1 步用 Z_{1p}在（此时的）最右边的一组$\overline{(x_1,\cdots,x_n)}$上计算 $g_1(x_1,\cdots,x_n)+1$，然后用 IR_1 将计算的结果（如果能停机的话）转移到带表达式的最左边。如果每一步都有结果的话，此时的带表达式为

$$\overline{(g_1(x_1,\cdots,x_n),\cdots,g_m(x_1,\cdots,x_n))}$$

第 m+2 步 最后，用 M 计算 $f(g_1(x_1,\cdots,x_n),\cdots,g_m(x_1,\cdots,x_n))$。

在上面的第 1——第 m+2 步的总共 $m+2$ 个步骤中，如果有

一个不能停机，则整个计算过程不能停止，$f(g_1(x_1,\cdots,x_n),\cdots,g_m(x_1,\cdots,x_n))$无定义。

定义 2.4.1 设 g 是一 n 元全函数，h 是一 $n+2$ 元全函数，如下定义的 $n+1$ 元全函数 f：

$$\begin{cases} f(x_1,\cdots,x_n,0)=g(x_1,\cdots,x_n) \\ f(x_1,\cdots,x_n,y+1)=h(x_1,\cdots,x_n,y,f(x_1,\cdots,x_n,y)) \end{cases}$$

称作由 g 和 h 经**原始递归**而得到的函数。

定理 2.4.2 设 f 是由 g 和 h 经原始递归而得到的函数，如果 g 和 h 都是图灵可计算的，则 f 也是图灵可计算的。

证 设 Z 是计算 $h(x_1,\cdots,x_n,y,z)+1$ 的正则图灵机，M 是计算 $g(x_1,\cdots,x_n)+1$ 的正则图灵机，我们来构造计算 f 的图灵机。其基本思想如下：

(1) 输入$(x_1,\cdots,x_n,\ y)$之后，先使用右 $n+1$ 减复制机 CR_{n+1}^- 将带表达式变为

$$\overline{(x_1,\cdots,x_n,y,x_1,\cdots,x_n,y-1,\cdots,x_1,\cdots,x_n,0,x_1,\cdots,x_n)}。$$

然后再用左 $n+1$ 消去机 ML_{n+1}，抹掉左边的 $n+1$ 个分量，得到

$$\overline{(x_1,\cdots,x_n,y-1,\cdots,x_1,\cdots,x_n,0,x_1,\cdots,x_n)}。$$

(2) 启动 M_p，算得终止的带表达式

$$\overline{(x_1,\cdots,x_n,y-1,\cdots,x_1,\cdots,x_n,0,f(x_1,\cdots,x_n,0))}$$

（注意：$f(x_1,\cdots,\ x_n,\ 0)=g(x_1,\cdots,\ x_n)$）。

(3) 反复 y 次使用 Z_p，依次得到带表达式

$$\overline{(x_1,\cdots,x_n,y-1,\cdots,x_1,\cdots,x_n,1,f(x_1,\cdots,x_n,1)},$$

$$\overline{(x_1,\cdots,x_n,y-1,\cdots,x_1,\cdots,x_n,2,f(x_1,\cdots,x_n,2)},$$

…………

$$\overline{(x_1,\cdots,x_n,y-1,f(x_1,\cdots,x_n,y-1)},$$

$$\overline{(f(x_1,\cdots,x_n,y))}。$$

(4) 最后，再抹掉一个 1，就得到输出 $f(x_1,\cdots,x_n,y)$。

这里的新问题就在于(3)中的“反复 y 次使用 Z_p”。在定理 2.4.1 的证明中，我们也曾重复 $m-1$ 次使用 CR_n，但那里的 m 是个

常数，而这里的 y 是个自变量！

解决问题的方法是通过增加一些条件转移指令来改造 Z_p。由上面的分析可以看出，是否还需要继续使用 Z_p，取决于当时的带表达式是多元数组的带表达式还是是单个自然数的带表达式，换句话说，就是看带上的 1 之间是否还有 0。

我们来构造 Z_1，它由以下指令构成：

$$\left.\begin{array}{l} q_0\ 1\ R\ q_0 \\ q_0\ 0\ R\ q_1 \\ q_1\ 0\ L\ q_2 \\ q_2\ 0\ L\ q_2 \\ q_2\ 1\ 0\ q_3 \end{array}\right\}$$ （如果带表达式上的 1 之间没有 0，抹掉一个 1，停机）

$$\left.\begin{array}{l} q_1\ 1\ L\ q_4 \\ q_4\ 1\ L\ q_4 \\ q_4\ 0\ L\ q_5 \\ q_5\ 1\ L\ q_4 \\ q_5\ 0\ R\ q_6 \\ q_6\ 0\ R\ q_7 \\ q_k\ 1\ 1\ q_0 \end{array}\right\}$$ （如果带表达式上的 1 之间有 0，移至左端，转入 Z_p 的运算。Z_p 运行结束，再重复上面的过程）

其中 $k=\theta(Z_p^{(7)})$。

令 $Z'=Z_1\cup Z_p^{(7)}$，则

$$CR_{n+1}^- * ML_{n+1} * M_P * Z'$$

是计算 $f(x_1,\cdots,x_n,y)$ 的图灵机。

定义 2.4.2 设 $g(x_1,\cdots,x_n,y)$ 为 $n+1$ 元部分函数，满足下式的（唯一的）n 元（部分）函数 f：

$$f(x_1,\cdots,x_n)=\begin{cases} y & \text{如果对每个 } k<y,\ g(x_1,\cdots,x_n,k) \\ & \text{都有定义但都不等于 0，而} \\ & \qquad g(x_1,\cdots,x_n,y)=0 \\ \text{无定义} & \text{否则。} \end{cases}$$

称为由 g 经**取极小运算**（**μ-运算**）而得到的（部分）函数，记作

$$f(x_1,\cdots,x_n)=\mu y[g(x_1,\cdots,x_n,y)=0]$$

应当注意,即使 g 是全函数,经 μ 运算而得到的函数 f 也未必是全函数。例如对二元全函数 $g(x, y)=x+y$ 取极小而得到的函数

$$\mu y[x+y=0]$$

就是一个仅在 $x=0$ 时有定义的一元函数。

定理 2.4.3 如果 $g(x_1,\cdots,x_n,y)$是可计算函数,则

$$f(x_1,\cdots,x_n)=\mu y[g(x_1,\cdots,x_n,y)=0]$$

也是可计算函数

证 设 Z 为计算 $g(x_1,\cdots,x_n,y)+1$ 的正则图灵机,我们按以下三个步骤来构造计算 f 的图灵机:

1. 输入$(x_1,\cdots,x_n)$之后,先在右边增加一个分量 0,使带表达式变为

$$\overline{(x_1,\cdots,x_n,0)}。$$

这可以由下述正则图灵机 Z_1 完成:

Z_1:

$q_0\ 1\ R\ q_0$
$q_0\ 0\ R\ q_1$
$q_1\ 1\ R\ q_0$
$q_1\ 0\ 1\ q_2$ (右移,找到连续两个 0,将右边的 0 改为 1)

$q_2\ 1\ L\ q_2$
$q_2\ 0\ L\ q_3$
$q_3\ 1\ L\ q_2$
$q_3\ 0\ R\ q_4$
$q_4\ 0\ R\ q_5$ (左移,移至最左边的 1 上,进入新状态,停机)

2. 利用左 $n+1$ 增复制机 CL_{n+1}^{+},将带表达式变为

$$\overline{(x_1,\cdots,x_n,1,x_1,\cdots,x_n,0)}。$$

3. 使用 Z_p。如果不能停机,则 $f(x_1,\cdots,x_n)$无定义;如果能停

机，则终止的带表达式为

$$\overline{(x_1,\cdots,x_n,1,g(x_1,x_2,\cdots,x_n,0))}。$$

4. 检查上述带表达式的最后一个分量是否为 0(即最后的一串 1 是否只有一个)：

(1) 如果只有一个 1(即 $g(x_1,\cdots,x_n,0)=0$)，则已经可以得到 $f(x_1,\cdots,x_n)=0$，抹掉前 n 个分量(使用 ML_n)，再抹掉第 $n+2$ 个分量(使用 MR_1)，最后再在仅剩的第 $n+1$ 个分量中抹掉一个 1，停机。

(2) 如果不只一个 1(即 $g(x_1,\cdots,x_n,0)\downarrow\neq0$)，重复步骤 2 以下的过程。

步骤 4 可以由以下图灵机 Z_2 完成：

令 $a=\theta(CL_{n+1}^{+}*Z_p)$。

Z_2:

$q_a\ 1\ R\ q_a$ $q_a\ 0\ R\ q_{a+1}$ $q_{a+1}1\ R\ q_a$ $q_{a+1}0\ L\ q_{a+2}$ $q_{a+2}0\ L\ q_{a+2}$	(找到最右边的 1)
$q_{a+2}1\ L\ q_{a+3}$ $q_{a+3}0\ L\ q_{a+4}$	(看这个 1 左边是不是 0，如果是 0……)
$q_{a+4}1\ L\ q_{a+5}$ $q_{a+5}1\ L\ q_{a+5}$ $q_{a+5}0\ L\ q_{a+7}$ $q_{a+7}1\ L\ q_{a+5}$ $q_{a+7}0\ R\ q_{a+9}$ $q_{a+9}0\ R\ q_{a+11}$	(如果是 0，移至左端，转入 $ML_n*MR_1*\{q_0\ 1\ 0\ q_1\}$)

$$\left.\begin{array}{l} q_{a+3}1\ R\ q_{a+6} \\ q_{a+6}1\ 0\ q_{a+8} \\ q_{a+8}0\ L\ q_{a+6} \\ q_{a+6}0\ L\ q_{a+10} \\ q_{a+10}1\ L\ q_{a+10} \\ q_{a+10}0\ L\ q_{a+12} \\ q_{a+12}1\ L\ q_{a+10} \\ q_{a+12}0\ R\ q_{a+14} \\ q_{a+14}0\ R\ q_5 \end{array}\right\}$$

（如果是 1，即 $g(x_1,\cdots,x_n,y)\neq 0$，抹掉最后一个分量，移至左端，重新转入 $CL_{n+1}^{+} * Z_p$ 的运算）

综合以上四个步骤，图灵机

$$(Z_1 * CL_{n+1}^{-} * Z_p)\cup Z_2\cup(ML_n * MR_1 * \{q_0\ 1\ 0\ q_1\})^{(a+11)}$$

是计算 f 的图灵机。

有了上述三条定理，我们就有了证明函数的图灵可计算性的新方法，与之有关的内容将在下一章中介绍。

图灵机问世之后，又有许多变型图灵机被构造了出来。如有一个读写头和多个工作带的多带图灵机、有多个读写头和一个工作带的多头图灵机以及多头多带图灵机，等等。但它们与本章所讲的图灵机所能计算的函数是相同的，只是计算速度有所提高。但是计算速度并不是递归论所关心的内容。有关这些图灵机的理论可参看[3]。

习　　题

1. 构造计算以下函数的图灵机，从而证明它们是图灵可计算函数：

(1) $f(x)=k$（k 是常数）

(2) $x+k$（k 是常数）

(3) $2x$，$3x$，kx（k 是常数）

(4) $x!$

2. 构造计算以下函数的图灵机，从而证明它们是图灵可计算函数：

(1) xy

(2) $\left[\frac{x}{y}\right]$（其中[ ]表示取整数部分）

(3) x^y

3. 设 $f(x_1,\cdots,x_n)$是 n 元的图灵可计算函数，用构造图灵机的方法证明函数

$$f(x_1,\cdots,x_n)+1$$

也是图灵可计算函数。

4. 设 $1\leqslant j\leqslant n$，n,j 是两个固定的正整数，n 元函数

$$U_j^{(n)}(x_1,\cdots,x_n)=x_j$$

称为投影函数（特别地，$U_1^{(1)}(x)=x$ 就是恒同函数）。证明，对任何 n 和 j，$U_j^{(n)}(x_1,\cdots,x_n)$是图灵可计算函数。

5. 定义二元函数 $x\dot{-}y$（读作"x 真减 y"）为

$$x\dot{-}y=\begin{cases}x-y & \text{当 } x\geqslant y,\\ 0 & \text{当 } x<y\end{cases}$$

（$x\dot{-}y$ 就是 $x-y$ 的补全函数）。证明 $x\dot{-}y$ 是图灵可计算函数。

6. 设 P 为 n 元谓词，如果 P 的特征函数

$$C_P(x_1,\cdots,x_n)=\begin{cases}1 & \text{当 } P(x_1,\cdots,x_n)\text{为真},\\ 0 & \text{当 } P(x_1,\cdots,x_n)\text{为假}\end{cases}$$

是图灵可计算函数，则称 P 为**图灵可判定谓词**。

证明以下谓词都是图灵可判定的：

(1) $x=y$

(2) $x<y$

(3) $x\leqslant y$

(4) $x|y$（x 可以整除 y）

7. 设 A 为自然数集（$A\subseteq N$），如果 A 的特征函数

$$C_A(x)=\begin{cases}1 & \text{当 } x\in A,\\ 0 & \text{当 } x\notin A\end{cases}$$

是图灵可计算函数，则称 A 为**图灵可计算集**。证明以下集合都是图灵可计算的：

(1) 偶数集，

(2) 奇数集，

(3) 空集$\varnothing$，

(4) 全体自然数的集合 N。

第三章

部分递归函数

本章我们从函数的构造和运算的角度来刻画能行性。这一方向的工作是哥德尔开创、由哥德尔和克利尼等人完成的。

三十年代初，哥德尔在证明不完全性定理时定义了一类函数，如今称为原始递归函数。此后，哥德尔和克利尼又在原始递归函数的基础上引入取极小运算（μ 运算），形成了部分递归函数（μ 递归函数）。我们的讨论就从原始递归函数出发。

§ 3.1 原始递归函数

原始递归函数类是从一些极其简单的初始函数出发，反复使用复合和原始递归两种函数运算而生成的函数类。初始函数的选择有一定的随意性。一般文献中常以下述三组函数作为初始函数：

1. 零函数　$O(x)=0$；
2. 后继函数　$S(x)=x+1$；
3. 对每一对正整数 $j,n(1\leqslant j\leqslant n)$ 的投影函数

$U_j^{(n)}(x_1,\cdots,x_n)=x_j$。

由于第 3 组中包括了无穷多个函数，所以初始函数有无穷多个。我们已经知道初始函数都是图灵可计算函数（第二章 § 2.2 中的例 1 和例 2 及第二章的习题 4）。

定义 3.1.1　从初始函数出发，经有穷多次使用复合和原始递归运算而得到的函数称为**原始递归函数**。全体原始递归函数

所构成的函数类记作 $\mathscr{P}$ 。

由于初始函数都是全函数，而由全函数经复合和原始递归而得到的函数也都是全函数，所以原始递归函数都是全函数。于是根据 § 2.4 中的定理 2.4.1 和定理 2.4.2 知

$$\mathscr{P} \subseteq \mathscr{C}$$

（$\mathscr{C}$ 是全体图灵可计算函数所构成的类），但由于 $\mathscr{C}$ 中有定义域不全的函数，所以

$$\mathscr{P} \neq \mathscr{C}\text{。}$$

我们来看一些原始递归函数的例子。

例 1 设 k 为任意的常数，函数

$$f_1(x)=x+k$$

是原始递归函数。因为

$$f_1(x)=\underbrace{S(S(\cdots S}_{k\text{重复合}}(x\underbrace{)\cdots)}_{k\text{层括号}}\text{。}$$

例 2 设 k 为任意常数，函数

$$f_2(x)=k$$

是原始递归函数。因为 f_2 可以由 f_1 和零函数复合而得

$$f_2(x)=f_1(O(x))\text{。}$$

例 3 恒同函数 $g(x)=x$ 是原始递归函数。因为它就是初始函数 $U_1^{(1)}(x)$。

例 4 二元函数 $x+y$ 是原始递归函数。因为它可以由一元函数 $g(x)=x$（恒同函数）和三元函数 $h(x,y,z)=S(z)$（一元的后继函数可以看成是三元函数的特例）经原始递归运算得到：

$$\begin{cases} x+0=x \\ x+(y+1)=(x+y)+1\text{。} \end{cases}$$

例 5 二元函数 $x\cdot y$ 是原始递归函数。因为它可以由$g(x)=0$（零函数）和 $h(x,y,z)=z+x$（加函数看成是三元函数的一个特例）经原始递归运算得到：

$$\begin{cases} x\cdot 0=0 \\ x\cdot(y+1)=x\cdot y+x\text{。} \end{cases}$$

例 6 $x \dot{-} 1$ 和 $x \dot{-} y$[1]都是原始递归函数。因为 $x \dot{-} 1$ 可以由零函数和恒同函数(看成三元函数)经原始递归得到：

$$\begin{cases} 0 \dot{-} 1 = 0 \\ (x+1) \dot{-} 1 = x; \end{cases}$$

$x \dot{-} y$ 可以由 $g(x)=x$ 和 $h(x,y,z)=z \dot{-} 1$ 经原始递归得到：

$$\begin{cases} x \dot{-} 0 = x \\ x \dot{-} (y+1) = (x \dot{-} y) \dot{-} 1。 \end{cases}$$

例 7 $|x-y|$是原始递归函数。因为它可以由 $x+y$ 和$x \dot{-} y$及 $y \dot{-} x$ 复合而得：

$$|x-y| = (x \dot{-} y) + (y \dot{-} x)。$$

例 8 符号函数

$$Sg(x) = \begin{cases} 1 & \text{当 } x>0, \\ 0 & \text{当 } x=0 \end{cases}$$

是原始递归函数。因为它可以由 $g(y)=0$ 和 $h(x,y,z)=1$ 经原始递归得到：

$$\begin{cases} Sg(0)=0 \\ Sg(x+1)=1。 \end{cases}$$

例 9 反符号函数

$$\overline{Sg}(x) = \begin{cases} 0 & \text{当 } x>0, \\ 1 & \text{当 } x=0 \end{cases}$$

是原始递归函数。因为

$$\overline{Sg}(x) = 1 \dot{-} Sg(x)。$$

例 10 二元函数

$$C_{=}(x,y) = \begin{cases} 1 & \text{当 } x=y, \\ 0 & \text{当 } x \neq y \end{cases}$$

(即等词的特征函数)是原始递归函数。因为

$$C_{=}(x,y) = \overline{Sg}(|x-y|)。$$

下面我们来建立原始递归函数的一些基本性质。

---

[1] 见第二章习题 5

定理 3.1.1　设 $f(x)$为一元的部分函数，二元函数

$$I_f(x,y)=\underbrace{f(f\ \cdots\ f}_{y+1\text{重复合}}(x\ \underbrace{)\ \cdots\ )}_{y+1\text{层括号}}$$

称为对 f 作**简单迭代**而得的函数(注意：y 是变量)。如果 $f(x)$是原始递归函数，则 $I_f(x,y)$也是原始递归函数。

证　取 $g(x)=f(x)$，$h(x,y,z)=f(z)$，则 $I_f(x,y)$可由 g,h 经原始递归而得：

$$\begin{cases}I_f(x,0)=f(x)\\ I_f(x,y+1)=f(I_f(x,y))。\end{cases}$$

定理 3.1.2　设 $f(x_1,\cdots,x_n,y)$为原始递归函数，则**有界和**

$$g(x_1,\cdots,x_n,y)=\sum_{k=0}^{y}f(x_1,\cdots,x_n,k)$$

和**有界积**

$$h(x_1,\cdots,x_n,y)=\prod_{k=0}^{y}f(x_1,\cdots,x_n,k)$$

都是原始递归函数。

证　g 可由 $f(x_1,\cdots,x_n,0)$和 $f(x_1,\cdots,x_n,y+1)+z$ 经原始递归得到：

$$\begin{cases}g(x_1,\cdots,x_n,0)=f(x_1,\cdots,x_n,0)\\ g(x_1,\cdots,x_n,y+1)=f(x_1,\cdots,x_n,y+1)+g(x_1,\cdots,x_n,y)；\end{cases}$$

h 可由 $f(x_1,\cdots,x_n,0)$和 $f(x_1,\cdots,x_n,y+1)\cdot z$ 经原始递归得到：

$$\begin{cases}h(x_1,\cdots,x_n,0)=f(x_1,\cdots,x_n,0)\\ h(x_1,\cdots,x_n,y+1)=f(x_1,\cdots,x_n,y+1)\cdot h(x_1,\cdots,x_n,y)。\end{cases}$$

定理 3.1.3　设 $g(x_1,\cdots,x_n,z)$为 $n+1$ 元全函数，如下定义的函数 f：

$$f(x_1,\cdots,x_n,y)=\begin{cases}b & \text{如果① } b\leqslant y\text{，且}\\ & \text{②}g(x_1,\cdots,x_n,b)=0\text{，且}\\ & \text{③对每个 } z<b\text{，} g(x_1,\cdots,x_n,z)\neq 0\\ y+1 & \text{否则}\end{cases}$$

称为对$g(x_1,\cdots,x_n,z)$**有界取极小(有界μ运算)**而得的函数,记作

$$f(x_1,\cdots,x_n,y)=\overset{y}{\mu}z[g(x_1,\cdots,x_n,z)=0]。$$

如果g是原始递归函数,则f也是原始递归函数。

证 容易验证

$$f(x_1,\cdots,x_n,y)=\sum_{z=0}^{y}[Sg(\prod_{k=0}^{z}g(x_1,\cdots,x_n,k))],$$

而等式右边的式子显然是原始递归的。

注意:有界取极小运算只能对全函数进行,这一点与取极小运算不同。

定理 3.1.4 设对每个i,$g_i(x_1,\cdots,x_n)(1\leqslant i\leqslant m+1)$都是原始递归函数,$\alpha_i(x_1,\cdots,x_n)(1\leqslant i\leqslant m)$也都是原始递归函数,并且$\alpha_i$中没有两个同时为0,则**分段定义**的函数$f$:

$$f(x_1,\cdots,x_n)=\begin{cases}g_1(x_1,\cdots,x_n) & 当\ \alpha_1(x_1,\cdots,x_n)=0\\ g_2(x_1,\cdots,x_n) & 当\ \alpha_2(x_1,\cdots,x_n)=0\\ \cdots\cdots & \cdots\cdots\\ g_m(x_1,\cdots,x_n) & 当\ \alpha_m(x_1,\cdots,x_n)=0\\ g_{m+1}(x_1,\cdots,x_n) & 其他情形。\end{cases}$$

是原始递归函数。

证 不难验证

$$f(x_1,\cdots,x_n)=\sum_{i=1}^{m}(g_i(x_1,\cdots,x_n)\cdot\overline{Sg}(\alpha_i(x_1,\cdots,x_n))$$

$$+g_{m+1}(x_1,\cdots,x_n)\cdot Sg(\prod_{i=1}^{m}\alpha_i(x_1,\cdots,x_n)),$$

等式右端的表达式显然是原始递归的。

定义 3.1.2 设P是n元谓词(n元关系),如果P的特征函数

$$C_P(x_1,\cdots,x_n)=\begin{cases}1 & 当\ P(x_1,\cdots,x_n)为真,\\ 0 & 当\ P(x_1,\cdots,x_n)为假\end{cases}$$

是原始递归函数,则称P为**原始递归谓词**。

定义 3.1.3 设$A\subseteq N^n$(即A是自然数n元组集合,$n\geqslant 1$),如

果 A 的特征函数

$$C_A(x_1,\cdots,x_n)=\begin{cases}1 & \text{当}(x_1,\cdots,x_n)\in A,\\ 0 & \text{当}(x_1,\cdots,x_n)\notin A\end{cases}$$

是原始递归函数，则称 A 为**原始递归集**。

由例 10，我们知道等词 = 是原始递归谓词。

例 11　设 $f(x_1,\cdots,x_n)$ 是原始递归函数，b 是常数，则谓词

$$f(x_1,\cdots,x_n)=b$$

是原始递归谓词。因为它的特征函数是

$$C_=(f(x_1,\cdots,x_n),b)。$$

例 12　空集 $\varnothing$ 是原始递归集。因为

$$C_\varnothing(x)=O(x)$$

全体自然数的集合 $N=\{0,1,2,3,\cdots\}$ 是原始递归的。因为

$$C_N(x)=1$$

例 13　单元集是原始递归集。设 $A=\{a\}$，则

$$C_A(x)=\begin{cases}1 & \text{如果}|x-a|=0,\\ 0 & \text{否则。}\end{cases}$$

因为由例 7 知，$|x-a|$ 是原始递归函数，于是根据定理 3.1.4，C_A 是原始递归函数。

例 14　$x<y$ 是原始递归谓词。因为

$$C_<(x,y)=Sg(y\dot{-}x)$$

定理 3.1.5　设 $f(x_1,\cdots,x_n)$ 和 $g(x_1,\cdots,x_n)$ 都是原始递归函数，则 n 元谓词

$$f(x_1,\cdots,x_n)=g(x_1,\cdots,x_n)$$

是原始递归谓词。

证　容易验证，该谓词的特征函数就是

$$C_=(f(x_1,\cdots,x_n),g(x_1,\cdots,x_n))$$

它当然是原始递归函数。

定理 3.1.6　设 P 是 $n+1$ 元原始递归谓词，对 P 作有界取极小运算而得到的函数

$$f(x_1,\cdots,x_n,y)=\begin{cases} b & \text{如果① } b\leqslant y\text{,且} \\ & \text{② } P(x_1,\cdots,x_n,b)\text{真,且} \\ & \text{③ 对每个 } z<b,\ P(x_1,\cdots,x_n,z)\text{假;} \\ y+1 & \text{否则} \end{cases}$$

(也记作 $f(x_1,\cdots,x_n,y)=\overset{y}{\mu}z[P(x_1,\cdots,x_n,z)]$) 也是原始递归函数。

证 容易验证

$$\overset{y}{\mu}zP(x_1,\cdots,x_n,z)=\overset{y}{\mu}z[\overline{Sg}(C_P(x_1,\cdots,x_n,z))=0],$$

根据定理 3.1.3,它是原始递归函数。

定理 3.1.7 若 P,Q 都是(n 元)原始递归谓词,则 $\neg P$, $P\wedge Q$, $P\vee Q$ 都是原始递归谓词。

证

$$C_{\neg P}(x_1,\cdots,x_n)=1\dot{-}C_P(x_1,\cdots,x_n),$$

$$C_{P\wedge Q}(x_1,\cdots,x_n)=C_P(x_1,\cdots,x_n)\cdot C_Q(x_1,\cdots,x_n),$$

$$C_{P\vee Q}(x_1,\cdots,x_n)=Sg(C_P(x_1,\cdots,x_n)+C_Q(x_1,\cdots,x_n))。$$

类似地,我们可以证明

定理 3.1.8 设 $A\subseteq N^n$, $B\subseteq N^n$,如果 A 和 B 都是原始递归集,则 $\overline{A}$, $A\cup B$, $A\cap B$ 也都是原始递归集。

推论 3.1.9 有穷集都是原始递归集(习题 4)。

我们引入**有界量词**符号$\overset{y}{\exists}$和$\overset{y}{\forall}$,其意义为

$$\overset{y}{\exists}zP(x_1,\cdots,x_n,z)\Leftrightarrow\exists z(z\leqslant y\wedge P(x_1,\cdots,x_n,z)),$$

$$\overset{y}{\forall}P(x_1,\cdots,x_n,z)\Leftrightarrow\forall z(z\leqslant y\rightarrow P(x_1,\cdots,x_n,z))。$$

定理3.1.10 如果 $P(x_1,\cdots,x_n,z)$是原始递归谓词,则

$$\overset{y}{\exists}zP(x_1,\cdots,x_n,z),$$

$$\overset{y}{\forall}zP(x_1,\cdots,x_n,z)$$

都是原始递归谓词。

证 记$\overset{y}{\exists}zP(x_1,\cdots,x_n,z)$为 $R(x_1,\cdots,x_n,y)$,

记$\overset{y}{\forall} zP(x_1,\cdots,x_n,z)$为$S(x_1,\cdots,x_n,y)$，则

$$C_R(x_1,\cdots,x_n,y)=Sg(\sum_{z=0}^{y}C_P(x_1,\cdots,x_n,z)),$$

$$C_S(x_1,\cdots,x_n,y)=\prod_{z=0}^{y}C_P(x_1,\cdots,x_n,z)。$$

最后，我们来讨论一个今后时常要用到的一个原始递归函数——**配对函数**。

考虑二元函数

$$J(x,y)=(\sum_{i=0}^{x+y}i)+x,$$

显然它是一个原始递归函数。容易证明，$J(x,y)$是从N^2到N的一一对应(所以称为配对函数)，这个对应关系可以用如下的无穷表格来表示：

J \ x y	0	1	2	3	4	5	6	7	8	9	…
0	0	1	3	6	10	15	21	28	36	45	…
1	2	4	7	11	16	22	29	37	46	56	…
2	5	8	12	17	23	30	38	47	57	…	…
3	9	13	18	24	31	39	48	58	…	…	…
4	14	19	25	32	40	49	59	…	…	…	…
5	20	26	33	41	50	60	…	…	…	…	…
6	27	34	42	51	61	…	…	…	…	…	…
7	35	43	52	62	…	…	…	…	…	…	…
8	44	53	63	…	…	…	…	…	…	…	…
9	54	64	…	…	…	…	…	…	…	…	…
……	…	…	…	…	…	…	…	…	…	…	…

表3.1　配对函数

由于$J(x,y)$是一一对应，所以等式

$$z=J(x,y)$$

也就确定了一个从 N 到 N^2 的一一对应。因此 x 和 y 也都是 z 的函数，我们把这两个函数分别记为

$$x=l(z) \quad 和 \quad y=r(z)。$$

于是

$$J(l(z),r(z))=z, \quad l(J(x,y))=x, \quad r(J(x,y))=y。$$

现在我们来证明 $x=l(z)$ 和 $y=r(z)$ 也都是原始递归函数。

注意到：

$$x\leqslant J(x,y), \qquad y\leqslant J(x,y),$$

于是

$$x=l(z)=\overset{z}{\mu}i[\overset{z}{\exists}j(J(i,j)=z)];$$

$$y=r(z)=\overset{z}{\mu}j[\overset{z}{\exists}i(J(i,j)=z)]。$$

由于 $J(i,j)=z$ 是原始递归谓词，加上有界量词后仍是原始递归谓词，再用有界取极小运算，得到的函数就是原始递归函数。

今后，我们把 $J(x,y)$，$l(z)$ 和 $r(z)$ 统称为配对函数，有时也把 $l(z)$ 和 $r(z)$ 叫做 $J(x,y)$ 的反函数。

实际上，象 $J(x,y)$ 这样的从 N^2 到 N 的原始递归的一一对应有无穷多个，都可以叫做配对函数。

类似地，我们可以定义从 N^n 到 N 的一一对应的原始递归函数及其反函数，称为 n 元配对函数。具体为：

$$J_1(x)=x,$$

$$J_2(x,y)=J(x,y),$$

$$J_3(x,y,z)=J(J(x,y),z),$$

$$\cdots\cdots\cdots\cdots$$

$$J_{n+1}(x_1,\cdots,x_n,x_{n+1})=J(J_n(x_1,\cdots,x_n),x_{n+1})。$$

它们当然也都有原始递归的反函数：令 $z=J_n(x_1,\cdots,x_n)$，则

$$x_1=\pi_1^{(n)}(z)=\underbrace{l(l\cdots l}_{n-1\text{层}}(z\underbrace{)\cdots)}_{n\text{层括号}},$$

$$x_2=\pi_2^{(n)}(z)=r(\underbrace{l\cdots l}_{n-2\text{层}}(z\underbrace{)\cdots)}_{n-1\text{层括号}},$$

············

$x_n = \pi_n^{(n)}(z) = r(z)$。

附记 事实上,把 $\pi_i^{(n)}(z)$看成 n, i, z 的三元函数(当 $i > n$ 和 $i = n = 0$时,定义 $\pi_i^{(n)}(z) = 0$)也是原始递归函数。证明留给读者。

§3.2 部分递归函数

定义3.2.1 由初始函数出发,经有穷次使用复合、原始递归和取极小三种运算而得到的函数,称为**部分递归函数**。全体部分递归函数的类记作 $\mathscr{R}$ 。

如果部分递归函数 f 的定义域是全的,就称之为**递归全函数**。

部分递归函数也称为 **μ-递归函数**,简称**递归函数**(请注意,在有些文献中是将"递归全函数"简称为"递归函数",本书中的"递归函数"则是指"部分递归函数")。

由定义3.2.1和上一节的结果,我们知道

$$\mathscr{P} \subseteq \mathscr{R} \subseteq \mathscr{C} 。$$

例1 $x - y$ 是递归函数。因为

$$x - y = \mu z[(z + y) \dot{-} x = 0]。$$

例2 处处无定义的函数 $f_\varnothing$是递归函数。因为它可以表示为

$$\mu z[z + 1 = 0]。$$

例3 $[\frac{x}{y}]$是递归函数。因为

$$\left[\frac{x}{y}\right] = \mu z[x < y \cdot z] - 1。$$

定理3.2.1 如果 $g(x_1, \cdots, x_n, z)$是递归函数,则有界和

$$\sum_{k=0}^{y} g(x_1, \cdots, x_n, k),$$

有界积

$$\prod_{k=0}^{y} g(x_1,\cdots,x_n,k)$$

和有界取极小

$$\mu_z^y[g(x_1,\cdots,x_n,k)=0]$$

所得到的函数也都是递归函数。

证明与定理3.1.2及定理3.1.3的证明相似，从略。

定理3.2.2　设对每个 $i(1\leqslant i\leqslant m)$，$g_i(x_1,\cdots,x_n)$和 $\alpha_i(x_1,\cdots,x_n)$都是递归函数，且 α_i 中没有两个同时为0，则分段定义的函数

$$f(x_1,\cdots,x_n)=\begin{cases} g_1(x_1,\cdots,x_n) & \text{当 } \alpha_1(x_1,\cdots,x_n)=0, \\ g_2(x_1,\cdots,x_n) & \text{当 } \alpha_2(x_1,\cdots,x_n)=0, \\ \cdots\cdots\cdots\cdots & \cdots\cdots\cdots\cdots \\ g_m(x_1,\cdots,x_n) & \text{当 } \alpha_m(x_1,\cdots,x_n)=0, \\ \text{无定义} & \text{其他情形。} \end{cases}$$

也是递归函数。

请注意，定理3.2.2不能照搬定理3.1.4的证明方法。这里有两点不同：一是本定理中的 α_i 只是部分递归函数，其定义域可能不全；二是对于“其他情形”不是定义为某个函数，而是无定义。因此如果我们仍采用定理3.1.4的方法把 f 表示成

$$f(x_1,\cdots,x_n)=\sum_{i=1}^{m} g_i\cdot\overline{Sg}(\alpha_i(x_1,\cdots,x_n)), \qquad (*)$$

那么，当 $\alpha_1(x_1,\cdots,x_n)$无定义，而 $\alpha_2(x_1,\cdots,x_n)$和 $g_2(x_1,\cdots,x_n)$都有定义且 $\alpha_2(x_1,\cdots,x_n)=0$时，按照 f 的定义应当有

$$f(x_1,\cdots,x_n)=g_2(x_1,\cdots,x_n),$$

而按照表达式(*)，却得出 $f(x_1,\cdots,x_n)$无定义的结论，因此(*)并不是 f 的表达式。

定理3.2.2的证明我们将在第五章给出。

由于今后我们要经常接触定义域不全的函数，为了避免误解，我们在此强调一下有关记法的约定：

今后，当我们写

$$f(x_1,\cdots,x_n)=a$$

的时候，我们的意思是$(x_1,\cdots,x_n)\in \mathrm{dom}\ f$并且函数值$f(x_1,\cdots,x_n)=a$。在需要特别明确这一点时，我们把这个意思写成

$$f(x_1,\cdots,x_n)\downarrow=a。$$

类似地，我们也把"$(x_1,\cdots,x_n)\in \mathrm{dom}\ f$但其函数值$f(x_1,\cdots,x_n)$不等于$a$"表示成

$$f(x_1,\cdots,x_n)\downarrow\neq a。$$

今后，除非明确知道$f(x_1,\cdots,x_n)$有定义（比如f是全函数），我们一般不使用$f(x_1,\cdots,x_n)\neq a$的记法。

由于原始递归函数都是全函数，而部分递归函数中有定义域不全的函数，因此由定义易知部分递归函数不都是原始递归函数。那么，部分递归函数中的全函数是否就都是原始递归函数呢？

阿克曼构造了一个函数（称为**阿克曼函数**）$A(x,y)$，并证明：$A(x,y)$是递归全函数，但不是原始递归函数。

阿克曼函数的定义是：

$$\begin{cases}A(0,y)=y+1,\\A(x+1,0)=A(x,1),\\A(x+1,y+1)=A(x,A(x+1,y))。\end{cases}$$

有关的证明可参看[3]。

定义3.2.2 设P是n元谓词，如果P的特征函数$C_P(x_1,\cdots,x_n)$是递归函数，则称P为**递归谓词**。

定义3.2.3 设$A\subseteq N^n$，如果A的特征函数$C_A(x_1,\cdots,x_n)$是递归函数，则称A为**递归集**。

原始递归集（谓词）都是递归的，但反之不然。

对于递归全函数$f(x_1,\cdots,x_n)$和$g(x_1,\cdots,x_n)$，谓词

$$f(x_1,\cdots,x_n)=g(x_1,\cdots,x_n)$$

是递归谓词。（证明留给读者。注意：全函数的条件是不可少的。）

以下几个定理给出了递归谓词和递归集的一些基本性质，它们的证明都与上一节中有关原始递归谓词和原始递归集的对应定

理相似，此处从略。

定理3.2.3　如果 P,Q 都是递归谓词，则 $\neg P$，$P\vee Q$ 和 $P\wedge Q$ 也都是递归谓词。

定理3.2.4　如果 $A\subseteq N^n, B\subseteq N^n, A, B$ 都是递归集，则 $\overline{A}$、$A\cup B$、$A\cap B$ 也都是递归集。

定理3.2.5　如果 $P(x_1,\cdots,x_n,z)$ 是 $n+1$ 元递归谓词，则加上有界量词后所得到的 n 元谓词

$$\overset{y}{\exists} zP(x_1,\cdots,x_n,z),$$

和

$$\overset{y}{\forall} zP(x_1,\cdots,x_n,z)$$

也都是递归谓词。

以后，为了方便，我们也对谓词作取极小运算：

$$\mu yP(x_1,\cdots,x_n,y)=\begin{cases} b & \text{如果① } P(x_1,\cdots,x_n,b)\text{为真，} \\ & \text{并且② 对每个 } z<b, \\ & \quad P(x_1,\cdots,x_n,z)\text{假；} \\ \text{无定义} & \text{否则。} \end{cases}$$

显然，

$$\mu yP(x_1,\cdots,x_n,y)=\mu y[\overline{S}g(C_P(x_1,\cdots,x_n,y))=0]。$$

所以，如果 P 是递归谓词，则 $\mu yP(x_1,\cdots,x_n,y)$ 是递归函数。

习　　题

1. 证明以下函数都是原始递归函数：

(1) $x!$　　　　(2) x^y（定义 $0^0=1$）

(3) $[\sqrt{x}]$　　　　(4) $p(x)=$第 $x+1$ 个素数，

2. 证明以下函数都是原始递归函数：

(1) $\mathrm{Max}(x,y)$（取 x，y 中较大的一个），

(2) $\mathrm{Min}(x,y)$（取 x，y 中较小的一个），

(3) $f(x,y)=\begin{cases} x,y \text{ 的最大公约数} & \text{当 } x\neq 0 \text{ 且 } y\neq 0 \\ 0 & \text{当 } x=0 \text{ 或 } y=0 \end{cases}$,

(4) $f(x,y)=[x,y]$(x,y 的最小公倍数),

3. 设 $g(x_1,\cdots,x_n,y)$, $\alpha(x_1,\cdots,x_n,y)$, $\beta(x_1,\cdots,x_n,y)$都是(原始)递归函数,证明以下四个函数都是(原始)递归的:

(1) $\sum\limits_{k=0}^{\alpha(x_1,\cdots,x_n,y)} g(x_1,\cdots,x_n,k)$,

(2) $\sum\limits_{k=\beta(x_1,\cdots,x_n,y)}^{\alpha(x_1,\cdots,x_n,y)} g(x_1,\cdots,x_n,k)$,

(3) $\prod\limits_{k=0}^{\alpha(x_1,\cdots,x_n,y)} g(x_1,\cdots,x_n,k)$,

(4) $\prod\limits_{k=\beta(x_1,\cdots,x_n,y)}^{\alpha(x_1,\cdots,x_n,y)} g(x_1,\cdots,x_n,k)$,

4. 证明以下谓词都是原始递归的(从而其外延集都是原始递归集):

(1) "x 是偶数", (2) "x 是素数",

(3) "x 是奇数", (4) "x 是2的方幂"。

5. 设 $A\subseteq N$,证明:如果 A 是有穷集,则 A 是(原始)递归集。

6. 定义二元谓词 div 为:

$$\mathrm{div}(x,y) \Leftrightarrow x\neq 0 \text{且} x \text{能整除} y$$

证明 $\mathrm{div}(x,y)$是原始递归谓词。

7. 证明函数

$$N(x)=\begin{cases} x \text{ 的因数的个数} & \text{当 } x\neq 0 \\ 0 & \text{当 } x=0 \end{cases}$$

是原始递归函数。

8. 证明函数

$$\zeta(x,y)=2^x(2y+1)-1$$

是一个配对函数。

9. 证明以下函数是部分递归函数:

(1) x^y (2) $[\frac{y}{x}]$

(3) rem(x,y)(y 被 x 除所得的余数)

10. 对任何自然数 x, x 可以唯一地表示成:

(1) $x=\sum_{i=0}^{\infty}\alpha_i 2^i$ (其中 α_i 为0或1),

如果 $x>0$, x 还可以唯一地表示成

(2) $x=2^{b_1}+2^{b_2}+\cdots+2^{b_l}$ (其中 $l\geqslant 1$, $0\leqslant b_1<b_2<\cdots<b_l$)

(3) $x=2^{a_1}+2^{a_1+a_2+1}+\cdots+2^{a_1+\cdots+a_k+(k-1)}$

定义:

$\alpha(i,x)=$(1)中的 α_i;

$$l^*(x)=\begin{cases}(2)\text{中的 } l & \text{如果 } x>0\\ 0 & \text{否则}\end{cases}$$

$$b(i,x)=\begin{cases}(2)\text{中的 } b_i & \text{如果 } x>0\text{且}1\leqslant i\leqslant l^*(x)\\ 0 & \text{否则}\end{cases}$$

$$a(i,x)=\begin{cases}(3)\text{中的 } a_i & \text{如果 } x>0\text{且}1\leqslant i\leqslant l^*(x)\\ 0 & \text{否则}\end{cases}$$

证明:α, l^*, b, a 都是(原始)递归函数。

11. 设 $g_1(x_1,\cdots,x_n)$, $g_2(x_1,\cdots,x_n)$, $\cdots$, $g_{m+1}(x_1,\cdots,x_n)$都是原始递归函数,$P_1(x_1,\cdots,x_n)$, $P_2(x_1,\cdots,x_n)$, $\cdots$, $P_m(x_1,\cdots,x_n)$都是原始递归谓词,并且 $P_1,\cdots,P_m$ 中没有两个同时为真,定义函数 f:

$$f(x_1,\cdots,x_n)=\begin{cases}g_1(x_1,\cdots,x_n) & \text{当 } P_1(x_1,\cdots,x_n)\text{为真}\\ \cdots\cdots\cdots\cdots & \\ g_m(x_1,\cdots,x_n) & \text{当 } P_m(x_1,\cdots,x_n)\text{为真}\\ g_{m+1}(x_1,\cdots,x_n) & \text{其他情形}\end{cases}$$

证明 f 是原始递归函数。

12. 设 $g_1(x_1,\cdots,x_n)$, $g_2(x_1,\cdots,x_n)$, $\cdots$, $g_m(x_1,\cdots,x_n)$都是部分递归函数,$P_1(x_1,\cdots,x_n)$, $P_2(x_1,\cdots,x_n)$, $\cdots$, $P_m(x_1,\cdots,x_n)$都是递归谓词,并且 $P_1,\cdots,P_m$ 中没有两个同时为真,定义函数 f:

$$f(x_1,\cdots,x_n)=\begin{cases} g_1(x_1,\cdots,x_n) & \text{当 } P_1(x_1,\cdots,x_n)\text{为真} \\ \cdots\cdots\cdots\cdots & \\ g_m(x_1,\cdots,x_n) & \text{当 } P_m(x_1,\cdots,x_n)\text{为真} \\ \text{无定义} & \text{其他情形} \end{cases}$$

证明 f 是部分递归函数。(注意比较11、12两题的差别。)

13. 设 $g(m,x_1,\cdots,x_n)$是递归全函数,证明 $n+1$元函数

$$f(m,x_1,\cdots,x_n)=J_{m+1}(g(1,x_1,\cdots,x_n),\cdots,g(m+1,x_1,\cdots,x_n))$$

是递归全函数。(本题中的“递归全函数”也可以改成“原始递归函数”。)

第四章

通用函数和通用图灵机

在前面两章我们分别定义了图灵可计算函数类 $\mathscr{C}$ 和部分递归函数类 R，并且知道

$$\mathscr{R} \subseteq \mathscr{C} 。$$

此时，十分自然的问题就是：是否也有

$$\mathscr{C} \subseteq \mathscr{R} ?$$

我们就从这一目标开始本章的讨论。我们将会发现所获得的结果要远远大于我们的期望。

§4.1 图灵机的算术化

我们的基本目标是证明图灵可计算函数都是部分递归函数。为此目的，我们先要将图灵机的符号语言转化为有关自然数的语言。将符号演算问题转化为自然数问题，是哥德尔在证明不完全性定理时最先使用的，因此通称为**哥德尔配数法**。具体实现这件工作的方法很多，我们采用配对函数来作这件事。

§4.1.1 图灵机的编号

从一般意义上讲，图灵机是可数集合（全体四元组的集合）的有穷子集，因此总共有可数无穷多部图灵机，我们可以通过一种能行的方法将它们一一列出，编上号码。由于这种编号方法具有重要意义，它可以将一切非自然数对象的判定问题（比如一个命题逻辑

的公式是否命题演算的定理)转化为有关自然数的判定问题，所以我们在这里讨论一种具体的编号方法。

记全体四元组的集合为Q，显然

$$Q=\{q_0,q_1,\cdots\}\times\{S_0,S_1,\cdots\}\times\{R,L,S_0,\cdots\}\times\{q_0,q_1,\cdots\}。$$

我们定义映射#：Q→N 为：

$$\#(\alpha)=\begin{cases}J_4(i,j,k+2,l) & \text{当 } \alpha \text{ 为 } q_i\,S_j\,S_k\,q_l, \\ J_4(i,j,0,l) & \text{当 } \alpha \text{ 为 } q_i\,S_j\,R\,q_l, \\ J_4(i,j,1,l) & \text{当 } \alpha \text{ 为 } q_i\,S_j\,L\,q_l\end{cases}$$

(其中 J_4 是四元配对函数，参看§3.1)。

容易看出，#是一个一一对应，而且是一个能行的一一对应：给定四元组 α 之后，我们可以使用四元配对函数实际算出 $\#(\alpha)$；反之，给定自然数 b，我们也可以实际找到满足 $\#(\alpha)=b$ 的那个(唯一的)四元组 α。$\#(\alpha)$称为四元组 α 的号码。

例如，当 $\alpha=q_0\,1\,0\,q_0$ 时，

$$\begin{aligned}\#(\alpha)&=J_4(0,1,2,0)=J(J_3(0,1,2),0)\\&=J(J(J(0,1),2),0)=J(J(1,2),0)\\&=J(7,0)=35。\end{aligned}$$

反之，设 $\#(\beta)=10$，我们可以算得

$$\pi_1^{(4)}(10)=l(l(l(10)))=0,$$
$$\pi_2^{(4)}(10)=r(l(l(10)))=0,$$
$$\pi_3^{(4)}(10)=r(l(10)))=0,$$
$$\pi_4^{(4)}(10)=r(10)=4,$$

于是

$$\beta=q_0\;0\;R\;q_4。$$

今后，我们将四元组 α 与 α 的号码 $\#(\alpha)$ 等同看待，当我们说四元组 b 时，意思是指号码为 b 的那个四元组。

使用类似的方法，我们也可以给图灵机编号。对于非空的图灵机 $Z=\{\alpha_1,\alpha_2,\cdots,\alpha_m\}$，定义 Z 的号码为

$$\#(Z)=J(J_m(\#(\alpha_1),\cdots\#(\alpha_m)),m)。$$

称 $\#(Z)$ 为 Z 的号码。这里应注意以下三点：

(1) 我们定义的图灵机是(四元组的)无序集，而 $\#(Z)$ 的值却与计算 $\#(Z)$ 时四元组的排列次序有关，因而一部图灵机可能有不只一个号码。实际上，由 m 个四元组组成的非空的图灵机有 $m!$ 个不同的号码。

(2) 由于四元组的有穷子集不都是图灵机(图灵机要求不能有两个四元组头两个符号对应相同)，所以并非每个自然数都是非空图灵机的号码。对于那些不是非空图灵机号码的自然数 n(有无穷多个)，我们以 n 为空图灵机的号码。这样，每部图灵机都有至少一个号码，而每个自然数也都是(唯一的)一部图灵机的号码。

(3) 图灵机的编号方法并不是一个函数，它不是单值的，而且表达式

$$\#(Z)=J(J_m(\#(\alpha_1),\cdots\#(\alpha_m)),m)$$

并不是我们通常意义下的函数式，因为它没有固定的元数。

对于我们来说，重要的是上述编号方法是能行的：给定图灵机 Z，我们可以实际算出 $\#(Z)$(尽管不同的人算出的结果可能不同)；给定自然数 a，我们可以实际找到唯一的图灵机 Z 使 $\#(Z)=a$。为了我们的目标，能作到这一步就够了。

通过上述编号方法，有关四元组或图灵机的问题就都转化为一个关于自然数的问题，今后我们将把它们等同看待。

如果两个四元组 α 和 β 的头两个符号不全对应相同，就称 α 和 β 是**相容**的。当 $a=\#(\alpha)$，$b=\#(\beta)$ 时，说

四元组 a 和 b 相容

就是说

四元组 α 和 β 相容

下面我们来讨论两个与四元组、图灵机有关的谓词：

例1 谓词"四元组 a 和 b 相容"是原始递归的。因为它等值于

$$(\pi_1^{(4)}(a)\neq\pi_1^{(4)}(b))\vee(\pi_2^{(4)}(a)\neq\pi_2^{(4)}(b))。$$

例2 谓词"c 是非空图灵机"是原始递归的。因为它等值于

$\overset{c}{\exists} m\{(m=r(c))\wedge \overset{m}{\forall} i \overset{m}{\forall} j((i=j)\vee(\pi_i^{(4)}(l(c))$与$\pi_j^{(4)}(l(c))$相容$))\}$。

在第二章中，我们以$\Phi_Z^{(n)}$表示图灵机Z所计算的n元函数。现在我们有了编号，如果$e=\#(Z)$，今后我们更常使用

$$\Phi_e^{(n)} \quad \text{或者} \quad \{e\}^{(n)}$$

来表示Z所计算的n元函数（即第e号图灵机所计算的n元函数）。当$n=1$时，上标(n)可以略去不写。

这样，函数序列

$$\Phi_0^{(n)},\ \Phi_1^{(n)},\cdots,\ \Phi_e^{(n)},\qquad \cdots(*)$$

或（按另一种写法）

$$\{0\}^{(n)},\{1\}^{(n)},\cdots,\{e\}^{(n)},\cdots \qquad (**)$$

包括了全部的n元图灵可计算函数。特别重要的是：由于每个n元可计算函数都有无穷多个计算它的图灵机，所以每个图灵可计算函数都在序列$(*)$和$(**)$中出现无穷多次。

与上述记号相应，我们以$W_e^{(n)}$表示$\Phi_e^{(n)}$的定义域，以$E_e^{(n)}$表示$\Phi_e^{(n)}$的值域。同样，当$n=1$时，上标(n)可以略去不写。

§4.1.2 瞬时描述的编号

图灵机的瞬时描述是由至少一个字母和恰好一个内部状态所组成的有穷序列（内部状态不是最右边的符号）。我们用如下方法来给它们编号：

先给每个字母配上一个偶数

$$\#(S_j)=2j,\qquad (j=0,1,2,\cdots)$$

给每个内部状态配上一个奇数

$$\#(q_i)=2i+1。\qquad (i=0,1,2,\cdots)$$

然后我们就可以给每个瞬时描述

$$\alpha:\quad S_{a_1}\cdots S_{a_m}\, q_i\, S_j\, S_{b_1}\cdots S_{b_n}$$

配上一个自然数$\#(\alpha)$，称作α的号码：

$$\#(\alpha)=J(J_{|\alpha|}(2a_1,\cdots,2a_m,2i+1,2j,2b_1,\cdots 2b_n),|\alpha|)$$

(其中$|\alpha|=m+n+2$,是瞬时描述α的长度)。

显然,每个瞬时描述都有唯一的号码。然而,并非每个自然数都是瞬时描述的号码,不过重要的是

引理4.1.2.1 “k是瞬时描述(的号码)”是原始递归谓词。

证 “k是瞬时描述(的号码)”等值于

$$r(k)=n \qquad (\text{即}\ \alpha\ \text{的长度}\ |\alpha|=n) \qquad ①$$

并且

$$\pi_i^{(n)}(l(k)) \qquad (i=1,2,\cdots,n)$$

中恰有一个是奇数,而且这个奇数不是最右边的符号,这可以表述为

$$\overset{n}{\exists} i((0<i<n)\wedge \text{“}\pi_i^{(n)}(l(k))\text{是奇数”}\wedge \overset{n}{\forall} j((j=i)\vee \vee \text{“}\pi_j^{(n)}(l(k))\text{是偶数”}) \qquad ②$$

显然,①和②都是原始递归的,而

$$\text{“}k\text{是瞬时描述(的号码)”} \Leftrightarrow ①\wedge②,$$

所以,“k是瞬时描述(的号码)”是原始递归谓词。

今后,我们就将“号码为k的瞬时描述”与自然数k等同看待。

引理4.1.2.2 谓词“y是输入$(x_1,\cdots,x_n)$的初始的瞬时描述”是原始递归的。

证 y是输入$(x_1,\cdots,x_n)$初始的瞬时描述,就是说以y为号码的瞬时描述形如

$$q_0 1^{(x_1+1)} 0 1^{(x_2+1)} 0 \cdots\cdots 0 1^{(x_n+1)} \qquad (*)$$

这等值于

$$\text{“}y\text{是瞬时描述”} \qquad ③$$

并且

$$k=r(y)=\sum_{i=1}^{n} x_i+2n \qquad ((*)\text{的长度}) \qquad ④$$

并且

$$\pi_1^{(k)}(l(y))=1 \qquad ((*)\text{的第一个符号为}\ q_0)$$

并且

$$\left.\begin{array}{l}\pi^{(k)}_{x_1+3}(l(y))=0\\ \pi^{(k)}_{x_1+x_2+5}(l(y))=0\\ \cdots\cdots\cdots\cdots\\ \pi^{(k)}_{x_1+\cdots+x_n+2n-1}(l(y))=0\end{array}\right\}\left\{\begin{array}{l}\text{(*)的第 } x_1+3\text{个、}\\ \text{第 } x_1+x_2+5\text{个}\cdots\cdots\\ \text{第 } x_1+\cdots x_n+2n-1\text{个}\\ \text{符号为0}\end{array}\right\} \quad ⑤$$

并且

$$\overset{k}{\forall} j(j=1 \vee j=x_1+3 \vee \cdots \vee j=x_1+\cdots+x_n+2n-1$$

$$\vee\, \pi^{(k)}_j(l(y))=2) \qquad \text{((*)中其余符号为1)} \qquad ⑥$$

由于③—⑥都是原始递归的，所以谓词"y 是输入$(x_1,\cdots,x_n)$的初始的瞬时描述"是原始递归的。

引理4.1.2.3　谓词"y 是相对于图灵机 e 的终止的瞬时描述"是原始递归的。

证　"y 是相对于图灵机 e 的终止的瞬时描述"是说 y 中的内部状态为 q_i，所对的方格中写有 S_j，而 e 中没有以 q_iS_j 开头的四元组。这等值于

$$k=r(y) \quad (y\text{的长度为 }k) \qquad ⑦$$

并且

$$m=r(e) \quad (e\text{中有 }m\text{ 个四元组}) \qquad ⑧$$

并且

$$\overset{k}{\exists} p \overset{k}{\exists} i \overset{k}{\exists} j\{\pi^{(k)}_p(l(y))=2i+1 \wedge \pi^{(k)}_{p+1}(l(y))=2j \wedge$$

$$\wedge \overset{m}{\forall} s(\pi^{(4)}_1(\pi^{(m)}_s(l(e))\neq i \vee \pi^{(4)}_2(\pi^{(m)}_s(l(e))\neq j))\} \qquad ⑨$$

由于⑦—⑨都是原始递归谓词，所以"y 是相对于图灵机 e 的终止的瞬时描述"是原始递归的。

在给瞬时描述编号的基础上，我们可以进一步给瞬时描述(有穷)序列编号。

设 Ψ 是瞬时描述序列

$$\Psi=\langle\Psi_1,\Psi_2,\cdots,\Psi_n\rangle,$$

定义 Ψ 的号码 $\#(\Psi)$ 为：

$$\#(\Psi)=J(J_n(\#(\Psi_1),\cdots,\#(\Psi_n)),n)。$$

显然，每个瞬时描述序列都有唯一的号码，但不是每个自然数都是瞬时描述序列的号码。对此我们仍然有

引理4.1.2.4　谓词“y是瞬时描述序列(的号码)”是原始递归的。

证　“y是瞬时描述序列(的号码)”等值于

$$r(y)=n \quad ⑩$$

并且

$$\overset{n}{\forall} i(\text{“}\pi_i^{(n)}(l(y))\text{是瞬时描述”}) \quad ⑪$$

显然，⑩和⑪都是原始递归的。

§4.1.3　图灵机的算术化

有了以上的编号方法，我们就可以将有关图灵机的一系列谓词转化为数论谓词。

例如，关于图灵机的谓词

$$\alpha \to \beta\ (Z)$$

就对应着一个数论谓词

$$y_1 \to y_2(e)$$

其意义为：“瞬时描述y_1经图灵机e的(一步)运算变为瞬时描述y_2”。

引理4.1.3.1　谓词$y_1 \to y_2(e)$是原始递归的。

证　图灵机的运算有三种，在此我们只讨论根据四元组$q_i\,S_j\,S_k\,q_l$所进行的运算，其他两种情形是类似的，留给读者。

如果$\alpha \to \beta\ (Z)$是根据Z中的指令$q_i\,S_j\,S_k\,q_l$进行的，则

$$\alpha:\ S_{a_1}\cdots S_{a_m}\,q_i\,S_j\,S_{b_1}\cdots S_{b_n}$$

$$\beta:\ S_{a_1}\cdots S_{a_m}\,q_l\,S_k\,S_{b_1}\cdots S_{b_n}$$

其中$\#(Z)=e$，$\#(\alpha)=y_1$，$\#(\beta)=y_2$。

在这种情形，谓词$y_1 \to y_2(e)$等值于

“y_1是瞬时描述”　①

并且

“y_2是瞬时描述” ②

并且

$r(y_1)=r(y_2)=M$ ③

并且

$r(e)=N$ ④

并且

“e 是非空图灵机” ⑤

并且

$\overset{M}{\exists} i \overset{M}{\exists} j \overset{M}{\exists} l \overset{M}{\exists} k \overset{M}{\exists} p \overset{N}{\exists} s\{\pi_p^{(M)}(l(y_1))=2i+1 \wedge \pi_{p+1}^{(M)}(l(y_1))=2j \wedge$
$\pi_p^{(M)}(l(y_2))=2l+1 \wedge \pi_{p+1}^{(M)}(l(y_2))=2k \wedge \pi_s^{(N)}(l(e))=$
$J_4(i,j,k+2,l) \wedge \overset{M}{\forall} u[u \neq p \wedge u \neq p+1 \rightarrow \pi_u^{(M)}(l(y_1))=$
$\pi_u^{(M)}(l(y_2))\}$ ⑥

显然谓词①—⑥都是原始递归的，因此其合取式也是原始递归的。于是在这种情形，谓词 $y_1 \rightarrow y_2(e)$是原始递归的。对一般情形，谓词

$y_1 \rightarrow y_2(e)$

是三个如上所说的合取式的析取(对应图灵机运算的三种情形)，因此是原始递归的。

对每个 $n \geqslant 1$，我们用

$T_n(x_1,\cdots,x_n,y,e)$

表示谓词“y 是图灵机 e 在输入 n 元数组$(x_1,\cdots,x_n)$后的计算”。这是一个 $n+2$元谓词。n 是参数，也就是说，对于每个 n，有一个这样的谓词，这一组谓词统称为**克利尼 T 谓词**。

引理4.1.3.2　$T_n(x_1,\cdots,x_n,y,e)$是原始递归谓词。

证　$T_n(x_1,\cdots,x_n,y,e)$等值于以下五个谓词的合取：

(1) y 是瞬时描述序列；

(2) $m=r(y)$；

(3) $\pi_1^{(m)}(l(y))$是输入$(x_1,\cdots,x_n)$的初始瞬时描述；

(4) $\overset{m}{\forall} i\{i=0 \vee (\pi_i^{(m)}(l(y)) \rightarrow \pi_{i+1}^{(m)}(l(y))\ (e))\}$；

(5) $\pi_m^{(m)}(l(y))$是相对于 e 的终止的瞬时描述。

我们已经证明(1)—(5)都是原始递归谓词(引理4.1.2.2—4.1.2.4,引理4.1.3.1),因此 $T_n(x_1,\cdots,x_n,y,e)$是原始递归谓词。

§4.2 通用函数和通用图灵机

对任意的 y,令

$m=r(y)$,

$a=\pi_m^{(m)}(l(y))$,

$k=r(a)$。

当 y 使上一节中 $T_n(x_1,\cdots,x_n,y,e)$为真时,m 就是该计算的步数,a 是终止的瞬时描述,k 是终止的瞬时描述的长度。

定义一元全函数 $U(y)$为:

$$U(y)=\sum_{i=1}^{k}(C_=(2,\pi_i^{(k)}(l(a))))$$

易见,当 $T_n(x_1,\cdots,x_n,y,e)$为真时,$U(y)$就是终止的瞬时描述中1的个数,也就是该计算的输出。根据定义的表达式,$U(y)$显然是原始递归函数。于是由上一节中的引理我们立即得到:

定理4.2.1 对每个 $n\geqslant 1$,

$$\Phi_e(x_1,\cdots,x_n)=U(\mu y T_n(x_1,\cdots,x_n,y,e)) \qquad (**)。$$

定理4.2.1是迄今为止我们所讨论到的最重要的定理,叫做**克利尼范式定理**。它向我们揭示了以下几点:

1. 图灵可计算函数都是部分递归函数,因为(**)中等式右边的表达式是部分递归的。与定理2.4.1—2.4.3及第二章习题4结合起来,就得到

$$\mathscr{R}=\mathscr{C}。$$

2. 以前,我们一直是将 $\Phi_e(x_1,\cdots,x_n)$中的 e 作为参数,现在,由定理4.2.1可以看出 e 也可以看作自变量,也就是说将 $\Phi_e(x_1,\cdots,x_n)$看作是 $n+1$元函数,它仍然是递归的,从而也是图灵可计

算的。

3. 为了得到全体部分递归函数，取极小运算是必需的。但由定理可以看出，为了得到任何一个具体的递归函数，取极小运算至多使用一次就可以，而且可以只限于对原始递归函数使用取极小运算。

4. 全体 n 元递归函数（可计算函数）有一个统一的表达式，它们都可以通过固定部分递归函数

$$U(\mu y T_n(x_1,\cdots,x_n,y,e)) \qquad (***)。$$

中的第 $n+1$ 个变元 e 的值而得到。

(***)式称为 n 元递归函数的**范式（克利尼范式）**该式所表示的函数（$n+1$元递归函数）称为 n 元递归函数的**通用函数**。

从图灵机的角度来讲，由于递归函数都是图灵可计算函数，所以存在一部图灵机 Z，Z 是计算(***)式的。也就是说，全体 n 元可计算函数（n 元递归函数）都可以使用同一部图灵机来计算，只是在计算时除了输入 n 个自变量的值之外还要输入一个代表该函数的参数 e。具有这种功能的图灵机 Z 称为**通用图灵机**。

通用函数和通用图灵机都不是唯一的。事实上，对每个 $n\geqslant 1$，都有无穷多个通用函数 $U(x_1,\cdots,x_n,y)$，它们都满足：

(1) 作为 $n+1$元函数，$U(x_1,\cdots,x_n,y)$是递归函数；

(2) 对任何 n 元递归函数 f，存在常数 k，使对任何 $x_1,\cdots,x_n$，

$$f(x_1,\cdots,x_n)=U(x_1,\cdots,x_n,k)。$$

这里的常数 k 称为递归函数 f 的（一个）指标。一个 n 元递归函数在不同的通用函数中可能有不同的指标。

利用类似的编号方法，我们也可以证明

定理4.2.2　对每个正整数 n，存在 $n+1$元递归全函数 $D(x_1,\cdots,x_n,y)$，使对每个 n 元原始递归函数 f，存在常数 k，对任何 $x_1,\cdots,x_n$，

$$f(x_1,\cdots,x_n)=D(x_1,\cdots,x_n,k)。$$

$n+1$元函数 $D(x_1,\cdots,x_n,y)$称为 n 元原始递归函数的通用函数。

出于篇幅的考虑，我们略去了定理4.2.2的证明，有兴趣的读者可以试着模仿本章的方法，对原始递归函数的生成过程编号，然后使用类似于证明定理4.2.1 的方法证明定理4.2.2。当然，这是一个相当烦琐的过程（参看[3]）。

到下一节我们会看到，定理4.2.2所断言存在的 n 元原始递归函数的通用函数 $D(x_1,\cdots,x_n,y)$ 作为一个 $n+1$ 元函数只是一个递归全函数，而不是原始递归函数。

§4.3 对角线方法

本节通过证明两个定理来介绍一下对角线方法。这个方法在可计算性理论、计算复杂性理论以及数理逻辑的其他分支中都占有十分重要的地位。

定理4.3.1　存在非递归的函数。

证　我们来实际构造一个一元的全函数 $f(x)$，并证明它不是递归函数。

我们已经知道全体的一元递归函数（图灵可计算函数）都包含在函数序列

$$\Phi_0,\ \Phi_1,\cdots,\ \Phi_e,\cdots \qquad ①$$

之中，于是可以作成一个无穷表：

Φ_x \ y / x	0	1	2	3	4	$\cdots$
0	$\Phi_0(0)$	$\Phi_0(1)$	$\Phi_0(2)$	$\Phi_0(3)$	$\Phi_0(4)$	$\cdots$
1	$\Phi_1(0)$	$\Phi_1(1)$	$\Phi_1(2)$	$\Phi_1(3)$	$\Phi_1(4)$	$\cdots$
2	$\Phi_2(0)$	$\Phi_2(1)$	$\Phi_2(2)$	$\Phi_2(3)$	$\Phi_2(4)$	$\cdots$
3	$\Phi_3(0)$	$\Phi_3(1)$	$\Phi_3(2)$	$\Phi_3(3)$	$\Phi_3(4)$	$\cdots$
4	$\Phi_4(0)$	$\Phi_4(1)$	$\Phi_4(2)$	$\Phi_4(3)$	$\Phi_4(4)$	$\cdots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	

考虑其对角线上的元素

$$\Phi_0(0),\ \Phi_1(1),\ \Phi_2(2),\ \cdots,\ \Phi_e(e),\ \cdots \qquad ②$$

它们实际上也定义了一个一元函数，我们将序列②逐项加以修改而得到一个一元函数 f：

$$f(x)=\begin{cases}\Phi_x(x)+1 & \text{如果 } \Phi_x(x) \text{ 有定义}\\ 0 & \text{如果 } \Phi_x(x) \text{ 无定义。}\end{cases}$$

如果 f 是递归函数，则 f 就是序列(*)中的某一个，比如是第 m 个，即 $f=\Phi_m$。于是对任何自然数 x，都有

$$f(x)=\Phi_m(x) \qquad ③$$

取 $x=m$，得到

$$f(m)=\Phi_m(m)。$$

但根据 f 的定义，若 $\Phi_m(m)$ 有定义，则

$$f(m)=\Phi_m(m)+1\neq\Phi_m(m);$$

若 $\Phi_m(m)$ 无定义，则

$$f(m)=0\neq\Phi_m(m)\uparrow。$$

就是说无论 $\Phi_m(m)$ 是否有定义，都有

$$f(m)\neq\Phi_m(m),$$

这与③矛盾。这个矛盾就表明 f 不是递归函数。

定理4.3.2　设 $D(x,y)$ 是一元原始递归函数的通用函数，则 $D(x,y)$ 不是二元原始递归函数。

证　由于 $D(x,y)$ 是一元原始递归函数的通用函数，所以全体一元原始递归函数都包括在函数序列

$$D(x,0),\ D(x,1),\ D(x,2),\ \cdots \qquad ④$$

之中。我们又可以做得如下的无穷表

D \ y / x	0	1	2	3	4	…
0	$D(0,0)$	$D(0,1)$	$D(0,2)$	$D(0,3)$	$D(0,4)$	…
1	$D(1,0)$	$D(1,1)$	$D(1,2)$	$D(1,3)$	$D(1,4)$	…
2	$D(2,0)$	$D(2,1)$	$D(2,2)$	$D(2,3)$	$D(2,4)$	…
3	$D(3,0)$	$D(3,1)$	$D(3,2)$	$D(3,3)$	$D(3,4)$	…
4	$D(4,0)$	$D(4,1)$	$D(4,2)$	$D(4,3)$	$D(4,4)$	…
⋮	⋮	⋮	⋮	⋮	⋮	

定义一元函数 g：

$$g(x)=D(x,x)+1 \quad ⑤$$

如果 $D(x,y)$ 是二元原始递归函数，则 g 当然也是原始递归函数，从而 g 是序列④中的某一个，不妨设是第 a 个，即对每个 x，都有

$$g(x)=D(x,a)。$$

令 $x=a$，得到

$$g(a)=D(a,a)。$$

而由⑤，我们又可以得到

$$g(a)=D(a,a)+1。$$

矛盾。这就表明 $D(x,y)$ 不是二元的原始递归函数（它是一个非原始递归的递归全函数）。

由定理4.3.2立即得到

推论4.3.3　存在二元全函数 $h(x,y)$，对每个固定的 y，$h(x,y)$ 作为 x 的一元函数都是原始递归的，但 $h(x,y)$ 作为 x 和 y 的二元函数却不是原始递归的。

对于递归全函数也有与推论4.3.3类似的结果（见习题8）

上面的定理4.3.1和定理4.3.2都是针对一元函数陈述的，事实上对任何的正整数 n，都有同样的结果，即：对任何正整数 n，都有非递归的 n 元函数；对任何的正整数 n，n 元原始递归函数的通用函数都不是原始递归的。

定理4.3.1和定理4.3.2的证明方法叫做**对角线方法**。当给定了一个函数序列 $g_0,g_1,\cdots$ 之后，我们就可以像上述两个证明那样作出无穷的表格，然后将对角线上的函数值逐个予以改变，得到一个新函数 f，这个对角线上的元素就提供了 $f\neq g_i(i=0,1,2,\cdots)$ 的证据，从而证明了函数 f 不在函数序列 $g_0,g_1,\cdots$ 之中。这个方法在数理逻辑的各个分支中有着广泛的应用。

习　题

1. 设 $\alpha_1 = q_0\ 1\ 0\ q_1$，$\alpha_2 = q_1\ 0\ R\ q_0$，计算 $\#(\alpha_1)$ 和 $\#(\alpha_2)$。

2. 已知 $\#(\alpha)=15$，求四元组 α。

3. 设图灵机 $Z=\{\alpha_1,\alpha_2\}$（见习题1），求 $\#(Z)$。

4. 已知瞬时描述 $\beta=q_0\ 1\ 0\ 1$，求 $\#(\beta)$。

5. 设 $\#(Z_1)=1$，$\#(Z_2)=6$，求图灵机 Z_1 和 Z_2。

6. 证明：谓词"x 是初始的瞬时描述"（不确定输入的元数）是原始递归的。

7. 设 $U(x_1,\cdots,x_n,y)$ 是一个 n 元通用函数，$g(x_1,\cdots,x_n,y)$ 是任意的 $n+1$ 元递归函数，则

$$\overline{U}(x_1,\cdots,x_n,y)=\begin{cases} g(x_1,\cdots,x_n,\frac{y-1}{2}) & \text{当 } y \text{ 为奇数} \\ U(x_1,\cdots,x_n,\frac{y}{2}) & \text{当 } y \text{ 为偶数。} \end{cases}$$

也是 n 元通用函数。

8. 用对角线方法证明：对全体一元递归全函数，不存在全的递归的通用函数，也就是说，如果有二元全函数 $g(x,y)$，使得对每个递归全函数 $f(x)$，都存在常数 a，满足

$$f(x)=g(x,a)$$

则 $g(x,y)$ 不是二元递归函数。

第五章

s-m-n 定理和递归定理

§5.1 丘奇论题

第一章中我们已经说过，在20世纪三十年代波斯特、丘奇、图灵、哥德尔、马尔科夫等人分别从不同的角度用不同的方法刻画了能行性概念，得到了波斯特系统可计算函数、λ-可定义函数、图灵可计算函数、部分递归函数、一般递归函数和马尔科夫算法可计算函数等等的概念。丘奇、图灵、马尔科夫三人在提出他们各自的理论之后都断言他们所定义的函数类就是直观意义上的能行可计算函数类。由于人们很快就证明了上述各种定义都是彼此等价的，于是便都接受了丘奇、图灵、马尔科夫的断言，把它称作**丘奇论题**或**丘奇-图灵论题**。在现代的递归论文献中，丘奇论题是如下陈述的：

丘奇论题 直观意义上的能行可计算函数的类恰好就是部分递归函数的类。

丘奇论题并不是严格意义下的数学命题或逻辑学命题，因而也就不存在在数学或逻辑学中严格证明它的问题。因为丘奇论题说及两个概念——能行可计算函数和部分递归函数，前者是一个没有严格定义的直观概念，后者则是一个有严格定义的数学概念。丘奇论题实际上就是断言我们对能行性所作的刻画是成功的，“部分递归函数”(以及与之等价的其他概念)恰如其份地刻画了直观

上的能行性概念。

丘奇论题的真理性要靠实践来检验。在20世纪的后六十多年中，特别是电子计算机问世之后，人们又提出了大量的理论计算模型，并且也都证明了它们仍与部分递归函数等价，这就更坚定了人们对丘奇论题的信心。如今在绝大多数的递归论文献中都接受丘奇论题作为一种证明手段：要证明某个函数（或谓词、集合）是递归的（可计算的、可判定的），只需实际给出一个计算（判定）它的算法就可以了。这样的证明称为**使用丘奇论题的证明**。

从现在起，在本书中如果没有特殊说明，都允许使用丘奇论题作为证明工具。我们来看几个例子。

例1 定义三元函数 $\Phi_{e,t}(x)$ 如下：

$$\Phi_{e,t}(x)=\begin{cases}1 & \text{如果对第 } e \text{ 号图灵机输入 } x \\ & \text{后在 } t \text{ 步运算之内能停机} \\ 0 & \text{否则。}\end{cases}$$

则 $\Phi_{e,t}(x)$ 是递归全函数。

我们可以使用如下的直观算法来计算 $\Phi_{e,t}(x)$：

给定 e,t,x 之后，根据图灵机的编号方法可以实际写出图灵机 Z，使 $\#(Z)=e$；然后输入 x，得到初始的瞬时描述 $q_0 1^{x+1}$；根据 Z 中的四元组进行运算，如果在 t 步之内已经停机，则 $\Phi_{e,t}(x)=1$；如果运算 t 步后仍未停机，则 $\Phi_{e,t}(x)=0$。

由于上述过程是能行的，所以根据丘奇论题 $\Phi_{e,t}(x)$ 是递归函数，而且根据定义它显然是全函数。

例2 定义三元函数 $f(x,y,z)$ 为

$$f(x,y,z)=\begin{cases}1 & \text{当 } z\in W_x\cup W_y, \\ \text{无定义} & \text{否则}\end{cases}$$

则 $f(x,y,z)$ 是递归函数（W_x 是 Φ_x 的定义域）。

我们可以使用如下的直观算法来计算 $f(x,y,z)$：

同时开动第 x 号图灵机和第 y 号图灵机，均输入 z。如果运算到某一步后有一部图灵机停机了，则 $f(x,y,z)=1$。显然，当

$f(x,y,z)$有定义时，$\Phi_x(z)$和$\Phi_y(z)$至少有一个有定义，因此如果$f(x,y,z)$有定义我们就总能算出$f(x,y,z)$的值。于是，根据丘奇论题，f是递归函数。

例3 定义函数$g(x)$为：

$$g(x)=\begin{cases}1 & \text{如果存在 } e \text{ 使 } \Phi_e(x)=1\\ \text{无定义} & \text{否则}\end{cases}$$

$g(x)$是递归函数。

计算$g(x)$的最直截了当的想法是：同时开动所有的图灵机，都输入x，如果其中有一部的输出是1，就可以知道$g(x)=1$；否则计算没有结局，也就是说$g(x)$没有定义。但是，图灵机总共有无穷多部，"同时开动所有的图灵机"不是一件能行的事情。为了保证能行性，我们可以分步进行：

第一步：计算 $\Phi_{1,1}(x)$

第二步：计算 $\Phi_{1,2}(x)$， $\Phi_{2,2}(x)$；

第三步：计算 $\Phi_{1,3}(x)$， $\Phi_{2,3}(x)$；

第四步：计算 $\Phi_{1,4}(x)$， $\Phi_{2,4}(x)$， $\Phi_{3,4}(x)$；

…………

第t步：计算 $\Phi_{1,t}(x)$， $\Phi_{2,t}(x)$， $\Phi_{3,t}(x)$， … $\Phi_{t,t}(x)$；

…………①

容易看出上面的过程就是每一步增加一部图灵机并且使已有的图灵机多运算一步。如果执行到某一步时有一部图灵机停机并输出1，则过程停止，$g(x)=1$，否则该过程将永无休止地进行下去（即$g(x)$无定义）。

根据丘奇论题，$g(x)$是递归函数。

例3 设$f(x_1,\cdots,x_n)$和$g(x_1,\cdots,x_n)$是两个递归函数，则函数

---

① 在上面的计算中没有Φ_0，不过这并不影响一般性，因为存在无穷多个k使$\Phi_k=\Phi_0$。

$$h(x_1,\cdots,x_n)=\begin{cases}1 & \text{如果 } f(x_1,\cdots,x_n) \text{ 和 } g(x_1,\cdots,x_n) \\ & \text{都有定义且} \\ & \quad f(x_1,\cdots,x_n)=g(x_1,\cdots,x_n) \\ \text{无定义} & \text{否则}\end{cases}$$

是递归函数。

我们可以用如下的算法计算 $h(x_1,\cdots,x_n)$：同时开动计算 f 和计算 g 的图灵机，都输入 $(x_1,\cdots,x_n)$，如果到某一步两部图灵机都停机并且输出相等，则 $h(x_1,\cdots,x_n)=1$；否则运算将永不停止，$h(x_1,\cdots,x_n)$ 无定义。因此，由丘奇论题知 h 是递归函数。

§5.2 s-m-n 定理

设 $f(x_1,\cdots,x_n,y_1,\cdots,y_m)$ 是 $n+m$ 元递归函数，由§4.1.1和§4.2知 f 必定在函数序列

$$\Phi_0^{(n+m)},\ \Phi_1^{(n+m)},\ \Phi_2^{(n+m)},\ \cdots \qquad (*)$$

中出现（无穷多次）。设 e 是 f 的一个指标，即

$$f=\Phi_e^{(n+m)}。$$

另一方面，我们也可以将 $y_1,\cdots,y_m$ 看作是参数，把 f 看成是关于 $x_1,\cdots,x_n$ 的 n 元函数，它（们）当然也是递归函数。于是，对于每一组具体的 $y_1,\cdots,y_n$，f 作为一个 n 元递归函数必定要在函数序列

$$\Phi_0^{(n)},\ \Phi_1^{(n)},\ \Phi_2^{(n)},\ \cdots \qquad (**)$$

中出现（无穷多次）。f 在(**)中的指标自然与 $y_1,\cdots,y_m$ 的值以及 f 在(*)中的指标有关。本节的第一个任务就是证明这种关系可以用一个 $m+1$元递归全函数表示。

定理5.2.1(s-m-n **定理**)　对每个 $m,n\geq1$，存在 $m+1$元递归全函数 $s_n^m(e,y_1,\cdots,y_m)$，使对任何 $x_1,\cdots,x_n$

$$\Phi_e^{(n+m)}(x_1,\cdots,x_n,y_1\cdots,y_m)=\Phi_{s_n^m(e,y_1,\cdots,y_m)}^{(n)}(x_1,\cdots x_n)$$

证　我们先从 $e,y_1,\cdots,y_m$ 出发构造图灵机 M，使得对任何

$x_1,\cdots,x_n$ 都有

$$\Phi_e^{(n+m)}(x_1,\cdots,x_n,y_1,\cdots,y_m,e)=\Phi_M^{(n)}(x_1,\ \cdots,x_n),$$

然后再证明 $e,y_1,\cdots,y_m$ 与 M 的号码 $\#(M)$ 之间的关系是一个 $m+1$ 元递归全函数。

给定 $e,y_1,\cdots,y_m$，设 Z 是图灵机，$\#(Z)=e$。根据 § 4. 1. 1的内容，这个 Z 是可以实际写出来的。

令下列四元组构成图灵机 M_1：

α_1	$q_0\ 1\ R\ q_0$	（读写头右移，找到连续两个0中的右边一个），
α_2	$q_0\ 0\ R\ q_1$	
α_3	$q_1\ 1\ R\ q_0$	
α_4	$q_1\ 0\ 1\ q_2$	（连续打印 y_1+1个1），
α_5	$q_2\ 1\ R\ q_2$	
α_6	$q_2\ 0\ R\ q_3$	
…………………		
α_{2y_1+4}	$q_{y_1+1}\ 0\ 1\ q_{y_1+2}$	
α_{2y_1+5}	$q_{y_1+2}\ 1\ R\ q_{y_1+2}$	
α_{2y_1+6}	$q_{y_1+2}\ 0\ R\ q_{y_1+3}$	（空一个0），
…………………		（连续打印 y_2+1个1），
…………………		（空一个0），
………………………………		
…………………		（连续打印 y_m+1个1），
$\alpha_{2\Sigma y_i+3m+1}$	$q_{\Sigma y_i+2m-1}\ 0\ 1\ q_{\Sigma y_i+2m}$	
$\alpha_{2\Sigma y_i+3m+2}$	$q_{\Sigma y_i+2m}\ 1\ L\ q_{\Sigma y_i+2m+1}$	（读写头移至左端）。
$\alpha_{2\Sigma y_i+3m+3}$	$q_{\Sigma y_i+2m+1}\ 1\ L\ q_{\Sigma y_i+2m+1}$	
$\alpha_{2\Sigma y_i+3m+4}$	$q_{\Sigma y_i+2m+1}\ 0\ L\ q_{\Sigma y_i+2m+2}$	
$\alpha_{2\Sigma y_i+3m+5}$	$q_{\Sigma y_i+2m+2}\ 1\ L\ q_{\Sigma y_i+2m+1}$	
$\alpha_{2\Sigma y_i+3m+6}$	$q_{\Sigma y_i+2m+2}\ 0\ R\ q_{\Sigma y_i+2m+3}$	
$\alpha_{2\Sigma y_i+3m+7}$	$q_{\Sigma y_i+2m+3}\ 0\ R\ q_{\Sigma y_i+2m+4}$	

M_1的作用是将瞬时描述

$$q_0\overline{(x_1,\cdots,x_n)}$$

变为

$$q_{\Sigma y_i+2m+4}\overline{(x_1,\cdots,x_n,y_1,\cdots,y_m)},$$

而且 M_1是正则的。令

$$M=M_1*Z,$$

容易看出

$$\Phi_Z^{(n+m)}(x_1,\cdots,x_n,y_1,\cdots,y_m)=\Phi_M^{(n)}(x_1,\cdots,x_n)。\qquad (*)$$

由于从 $e,y_1,\cdots,y_m$ 构造 M 的过程是能行的，所以令

$$s_n^m(e,y_1,\cdots,y_m)=\#(M)。$$

根据丘奇论题 $s_n^m(e,y_1,\cdots,y_m)$是递归函数，而由上面的过程和 s_n^m 的定义，s_n^m 显然是一个全函数。而由(*)可得

$$\Phi_e^{(n+m)}(x_1,\cdots,x_n,y_1\cdots,y_m)=\Phi^{(n)}_{s_n^m(e,y_1,\cdots,y_m)}(x_1,\cdots x_n)。$$

在定理的证明中我们使用了丘奇论题。当然，我们也可以不使用丘奇论题而直接用形式化的方法证明上述定理，而且我们还可以进一步证明定理中的 $s_n^m(e,y_1,\cdots,y_m)$实际上可以是原始递归函数。不过我们不在这里作这些事情了，今后在使用 s-m-n 定理时只需要 $s_n^m(e,y_1,\cdots,y_m)$是递归全函数就可以了。有关 s-m-n 定理的形式化证明可参看[2]。

推论5.2.2　设 $f(x_1,\cdots,x_n,y_1.\cdots,y_m)$是 $n+m$ 元递归函数，则存在 m 元递归全函数 $t_n^m(y_1,\cdots,y_m)$使

$$f(x_1,\cdots,x_n,y_1\cdots,y_m)=\Phi^{(n)}_{t_n^m(y_1,\cdots,y_m)}(x_1,\cdots x_n)。$$

证　由于 f 是递归函数，故有常数 e 使

$$f(x_1,\cdots,x_n,y_1\cdots,y_m)=\Phi_e(x_1,\cdots x_n,y_1,\cdots,y_m)。$$

根据定理5.2.1知有递归全函数 $s_n^m(e,y_1,\cdots,y_m)$使

$$f(x_1,\cdots,x_n,y_1\cdots,y_m)=\Phi^{(n)}_{s_n^m(e,y_1,\cdots,y_m)}(x_1,\cdots x_n),$$

由于此处 e 是常数，令

$$t_n^m(y_1,\cdots,y_m)=s_n^m(e,y_1,\cdots,y_m),$$

即得

$$f(x_1,\cdots,x_n,y_1\cdots,y_m)=\Phi^{(n)}_{t_n^m(y_1,\cdots,y_m)}(x_1,\cdots x_n)。$$

从定理5.2.1到推论5.2.2的过程看上去是相当自然的，但二者在构造性方面却很不相同。定理5.3.1的证明完全是构造性的——从 $e,y_1,\cdots,y_m$ 的值我们可以实际地算出 $s_n^m(e,y_1,\cdots,y_m)$ 的函数值，因此 s_n^m 这个函数对我们来说就不仅仅是存在，而且是可以实际计算的。但推论的证明中有一步是非构造性的，即从 f 找到它的一个指标 e 这一步。由于递归论中并不限制定义函数的方法，所以对于有些函数我们虽然可以（用非构造性的方法）证明它的递归性，但却不见得能找到它的指标。

例如如下定义的函数 $h(x)$：

$$h(x)=\begin{cases}1 & \text{如果 } \pi \text{ 的十进展开式中有连续 } x \text{ 个5} \\ 0 & \text{否则}\end{cases}$$

由于我们知道 $\pi=3.1415926535\cdots$，因此可以得到

$$h(0)=h(1)=1。$$

此外，由 h 的定义容易看出对任何的 x，

如果 $h(x+1)=1$，则 $h(x)=1$

（如果有连续 $x+1$ 个5，自然就有连续 x 个5）。

因此，函数 h 只有两种可能：或者是常函数1（即对任何 x，$h(x)=1$），或者存在一个 k，使对任何 $x\leqslant k$，$h(x)=1$；而对任何 $x>k$，$h(x)=0$。无论是哪种情形，h 都是递归函数，因此我们可以证明 h 是递归函数，但由于我们并不确切知道 h 究竟是哪种情形，所以我们并不知道 h 的指标。

如果推论中的 f 是像 h 这类非构造性定义的函数，我们就无法实际算出推论中所断言存在的那个函数 t_n^m 的值，于是这个 $t_n^m(y_1,\cdots,y_m)$ 对我们而言就仅仅是存在，而不能实际计算它的值。

§5.3 递归定理

s-m-n 定理在递归论中有有很多重要的应用，其中一个典型的例子就是证明递归定理。

递归定理也叫**第二递归定理**或**不动点定理**，是递归论研究中的一个很重要的定理。

定理5.3.1（**递归定理**） 设 f 是任意的一元递归全函数，存在自然数 n，使

$$\Phi_{f(n)}=\Phi_n。$$

（n 称为 f 的不动点。）

证 由范式定理（定理4.2.1），我们知道

$$\Phi_{f(\Phi_y(y))}(x)$$

是二元递归函数。于是根据推论5.2.2，有一元递归全函数 $t(y)$，使得对任何 x，

$$\Phi_{f(\Phi_y(y))}(x)=\Phi_{t(y)}(x), \qquad ①$$

也就是

$$\Phi_{f(\Phi_y(y))}=\Phi_{t(y)}。 \qquad ②$$

既然 t 是一元递归函数，于是就有常数 m 使 $t=\Phi_m$。从而由②可以得到

$$\Phi_{f(\Phi_y(y))}=\Phi_{\Phi_m(y)}。 \qquad ③$$

在③式中取 $y=m$，$n=\Phi_m(m)$（由于 $t(y)$ 是全函数，故 $\Phi_m(m)=t(m)$ 有定义），就得到

$$\Phi_{f(n)}=\Phi_n。$$

由定理5.3.1立即可以得到

推论5.3.2 对任何递归全函数 $f(x)$，存在自然数 n 使

$$W_n=W_{f(n)}。$$

递归定理还有一个推广的形式：

定理5.3.3(**带参数的递归定理**)　设 $f(x,y)$是任意的二元递归全函数，存在递归全函数 $n(y)$，使对任何 y

$$\Phi_{n(y)}=\Phi_{f(n(y)),y)}。$$

证　定义函数 $g(x,y,z)$如下：

$$g(x,y,z)=\begin{cases}\Phi_{\Phi_x(x,y)}(z) & \text{如果 } \Phi_x(x,y)\text{有定义,} \\ \text{无定义} & \text{否则。}\end{cases} \quad ①$$

由丘奇论题知 $g(x,y,z)$是递归函数。于是根据 s-m-n 定理，存在(二元)递归全函数 $d(x,y)$使

$$\Phi_{d(x,y)}(z)=g(x,y,z),$$

也就是

$$\Phi_{d(x,y)}(z)=\begin{cases}\Phi_{\Phi_x(x,y)}(z) & \text{如果 } \Phi_x(x,y)\text{有定义,} \\ \text{无定义} & \text{否则。}\end{cases}$$

$f(d(x,y),y)$当然是(二元的)递归函数,取它的一个指标 e(注意 e 是常数),即

$$\Phi_e(x,y)=f(d(x,y),y)。$$

令 $n(y)=d(e,y)$,则由①可得

$$\Phi_{d(e,y)}=\Phi_{\Phi_e(e,y)}=\Phi_{f(d(e,y),y)},$$

也就是

$$\Phi_{n(y)}=\Phi_{f(n(y)),y)}。$$

由定理5.3.3立即可以得到

推论5.3.4　对任何(二元)递归全函数 $f(x,y)$,存在递归全函数 $n(y)$使

$$W_{n(y)}=W_{f(n(y)),y)}。$$

递归定理还可以推广到更多元的情形,它们也可以通过配对函数还原为二元的情形,读者不妨自己试一下。

习　　题

1. 利用丘奇论题证明定理3.2.2。

2. 利用丘奇论题证明如下定义的函数

$$f(x,y,z)=\begin{cases}0 & \text{如果 } \Phi_z(y)\downarrow, \\ \uparrow & \text{否则}\end{cases}$$

是递归函数。

3. 利用丘奇论题证明如下定义的函数

$$f(x,y)=\begin{cases}1 & \text{如果 } W_x\neq\varnothing, \\ \uparrow & \text{否则}\end{cases}$$

是递归函数。

第六章

哥德尔不完全性定理

初等数论的形式系统 **PA** 是为了刻画自然数模型 $\mathscr{N}$ 而建立起来的。我们当然希望在这个标准模型中为真的句子都是 **PA** 的定理，即满足：

$$\text{如果 } \mathscr{N} \models \alpha\text{，则 } \mathbf{PA} \vdash \alpha\text{。} \qquad ①$$

如果①成立，则 **PA** 是完全的。

哥德尔不完全性定理就是说一阶形式算术理论 **PA** 不是完全的。也就是说存在一个 **PA** 句子(闭公式)使得

$$\mathscr{N} \models \alpha \quad \text{但} \quad \mathbf{PA} \nvdash \alpha\text{。} \qquad ②$$

由于任何一个句子在 $\mathscr{N}$ 中或者为真或者为假，因此如果 $\mathscr{N}$ 确实是**PA** 的模型(从而 **PA** 一致)，②就等价于说存在一个**PA** 句子 α 使得

$$\mathbf{PA} \nvdash \alpha \quad \text{且} \quad \mathbf{PA} \nvdash \neg\alpha\text{。} \qquad ③$$

下面我们就按③来讨论哥德尔不完全性定理。

§6.1 形 式 算 术

十九世纪八十年代，戴德金和皮亚诺先后给出了自然数公理。虽然戴德金的公理略早于皮亚诺，但不如皮亚诺的公理简单明了，因而后来广泛流传的是皮亚诺的公理。在现代文献中，皮亚诺公理一般是如下陈述的：

Ⅰ 0是(自然)数；

Ⅱ 每个(自然)数都有一个后继数;

Ⅲ 不同(自然)数的后继也不同;

Ⅳ 0不是任何(自然)数的后继数;

Ⅴ 数学归纳法。

所谓形式算术,就是以一阶逻辑的形式语言陈述皮亚诺公理而得到的形式理论,因而也称作"皮亚诺算术",简记作 **PA** (Peano's Arithmatic)。在不同的文献中对 **PA** 的陈述方式也有所不同,这主要是出于各人的偏好,并没有什么实质区别。本书采用比较通用的方式。

PA 由以下内容组成:

一. 形式语言 $\mathscr{L}$:

1. 初始符号

(1) 可数无穷多个个体变元: v_1, v_2, v_3,…

(以后用小写拉丁字母 x, y, z, …(可以加上、下标)表示任何个体变元。)

(2) 逻辑联接词: ¬(并非), →(如果…那么…);

(3) 量词: ∀(任何);

(4) 等词: = ;

(5) 常项: **0**;

(6) 一元函数符号: S(后继函数);

(7) 二元函数符号: +, ·;

(8) 辅助记号: 括号。

2. 项(以下用黑体小写拉丁字母 $\mathbf{r}$, $\mathbf{s}$, $\mathbf{t}$, … (可以加上、下标)表示任何项)

(1) 单个变元是项;

(2) **0** 是项;

(3) 如果 $\mathbf{s}$,$\mathbf{t}$ 是项,则 $S(\mathbf{t})$, $(\mathbf{s}+\mathbf{t})$, $(\mathbf{s}\cdot\mathbf{t})$ 也都是项;

如果一个项中不含个体变元(例如 0, $S(\mathbf{0})$, $(\mathbf{0}+S(\mathbf{0}))$ 等等)就称之为**无变元项**。在无变元项中, 0, $S(\mathbf{0})$, $S(S(\mathbf{0}))$, …有

着特别重要的地位，我们称它们为**数字**。事实上，它们正是自然数0，1，2，…的形式表达式。

今后，对于自然数 x，我们用 $\bar{x}$ 表示 x 的数字即自然数 x 的形式表达式。例如 $\bar{1}=S(\mathbf{0})$，$\bar{2}=S(S(\mathbf{0}))$，$\bar{3}=S(S(S(\mathbf{0})))$，等等。

3. 公式（以下用小写希腊字母 $\alpha,\beta,\gamma,\cdots$（可以加上、下标）表示任何公式）

(1) 如果 s,t 是项，$(s=t)$ 是公式（称为原子公式）；

(2) 如果 α,β 是公式，则 $\neg\alpha$，$(\alpha\rightarrow\beta)$ 也是公式；

(3) 如果 α 是公式，x 是变元，则 $\forall x\alpha$ 是公式。

4. 基本定义

(1) $(\alpha\vee\beta)=_{df}(\neg\alpha\rightarrow\beta)$；

(2) $(\alpha\wedge\beta)=_{df}\neg(\alpha\rightarrow\neg\beta)$；

(3) $(\alpha\leftrightarrow\beta)=_{df}((\alpha\rightarrow\beta)\wedge(\beta\rightarrow\alpha))$；

(4) $\exists x\alpha=_{df}\neg\forall x\neg\alpha$；

(5) $s\neq t=_{df}\neg(s=t)$。

以后，为了书写简便，我们也采用大多数文献所使用的省略括号的约定。

二. **项和公式的复杂度** 项 t 的**复杂度** $\deg(t)$ 是 t 中函数符号（S，$+$，$\cdot$）的个数；公式 α 的**复杂度** $\deg(\alpha)$ 是 α 中 $\forall$、$\neg$ 和 $\rightarrow$ 的总个数。例如

$$\deg(v_1)=0,\qquad \deg(v_1+v_3)=1,$$
$$\deg(S(v_2)\cdot(v_1+v_3))=3,$$
$$\deg(v_1+v_3=S(\mathbf{0}))=0,$$
$$\deg(\forall v_1\neg(S(v_1)=\mathbf{0}))=2。$$

三. **逻辑公理（模式）**

$Ax1$ $\alpha\rightarrow\beta\rightarrow\alpha$；

$Ax2$ $(\alpha\rightarrow\beta\rightarrow\gamma)\rightarrow(\alpha\rightarrow\beta)\rightarrow\alpha\rightarrow\gamma$；

$Ax3$ $(\neg\alpha\rightarrow\beta)\rightarrow(\neg\alpha\rightarrow\neg\beta)\rightarrow\alpha$；

$Ax4$ $\forall x\alpha\rightarrow\alpha(x/t)$ （x 为任何项）；

$Ax5$　$\alpha \to \forall x\alpha$　（x 不在 α 中自由出现）；

$Ax6$　$\forall x(\alpha \to \beta) \to \forall x\alpha \to \forall x\beta$；

$Ax7$　$t=t$　（t 是任何项）；

$Ax8$　$s=s' \to t=t' \to s=t \to s'=t'$　（s，s'，t，t'是任何项）；

$Ax9$　$s=t \to S(s)=S(t)$　（s，t 是任何项）；

$Ax10$　$s=s' \to t=t' \to s+t=s'+t'$　（s，s'，t，t'是任何项）；

$Ax11$　$s=s' \to t=t' \to s \cdot t=s' \cdot t'$　（s，s'，t，t'是任何项）；

四．非逻辑公理（模式）

PA1　$0 \neq S(x)$；

PA2　$S(x)=S(y) \to x=y$；

PA3　$x+0=x$；

PA4　$x+S(y)=S(x+y)$；

PA5　$x \cdot 0=0$；

PA6　$x \cdot S(y)=(x \cdot y)+x$；

PA7（归纳公理模式）　对每个恰含 n＋1个自由变元的公式 $\varphi(x,x_1,\cdots,x_n)$，有公理

$$\varphi(0,x_1,\cdots,x_n) \wedge \forall x(\varphi(x,x_1,\cdots,x_n) \to \varphi(S(x),x_1,\cdots,x_n)) \\ \to \forall x\varphi(x,x_1,\cdots,x_n)；$$

五．变形规则

1．分离规则(m.p)：从 α 和 $\alpha \to \beta$ 得到 β；

2．$\forall$概括规则：对任何变元 x，从 α 得到$\forall x\alpha$。

下面我们来定义 PA 系统中的证明和演绎。

定义6.1.1　所谓 **PA 证明**指的是一个有穷的公式序列 φ_1，$\varphi_2,\cdots,\varphi_n$。对每个 $k(1 \leqslant k \leqslant n)$，$\varphi_k$ 或者是

（1）一条公理（可以是逻辑公理，也可以是非逻辑公理）；或者是

（2）由 φ_i 和 $\varphi_j(1 \leqslant i,j < k)$经 m.p 得到（此时 φ_j 形如 $\varphi_i \to \varphi_k$）或者是

（3）由 $\varphi_i(i < k)$经$\forall$概括得到的。

如果存在一个 PA 证明的最后一个公式是 α,则称 α 是 **PA 定理**,记作

$$\mathbf{PA}\vdash\alpha \qquad \text{或者} \qquad \vdash_{\mathbf{PA}}\alpha。$$

在不致引起混淆时,断定符 $\vdash$ 的下标 PA 可以省略。

定义6.1.2 设 Φ 是一 $\mathscr{L}$ 公式集。所谓从 Φ 作出的一个 **PA 演绎**指的是一个有穷的公式序列 $\varphi_1, \varphi_2, \cdots, \varphi_n$;对每个 $k(1\leqslant k\leqslant n)$, φ_k 或者是

(1) PA 定理,或者是

(2) $\varphi_k\in\Phi$,或者是

(3) 由 φ_i 和 $\varphi_j(1\leqslant i,j<k)$经 m.p 得到(此时 φ_j 形如 $\varphi_i\to\varphi_k$)。

如果存在一个从 Φ 作出的 PA 演绎的最后一个公式是 α,则称由 Φ 可以演绎出 α,记作

$$\Phi\vdash_{\mathbf{PA}}\alpha;$$

在不致引起混淆时,下标 PA 可以省略。容易证明

定理6.1.1(**演绎定理**)

$$\Phi\cup\{\beta\}\vdash_{\mathbf{PA}}\alpha \quad \text{当且仅当} \quad \Phi\vdash_{\mathbf{PA}}\beta\to\alpha。$$

我们可以在 PA 中表达平时所用到的各种有关自然数的命题。但应注意,许多有关自然数集的命题却不能在 PA 中表达。例如

每个非空的自然数集都有最小元。

如果 $A\cup B$ 是无穷集,则 A,B 至少有一个是无穷集。

都不能在 PA 中表达。它们实质上都是二阶语句,而 PA 是一阶的。

建立 PA 系统是为了刻画我们天天都在使用的自然数结构

$$\mathscr{N}=\langle \mathrm{N}, 0, S, +, \cdot\rangle。$$

(其中 N 是全体自然数所组成的集合:$\mathrm{N}=\{0,1,2,\cdots\}$;0是自然数零;$S$ 是自然数上的后继运算:$S(x)=x+1$;$+$ 和 $\cdot$ 分别是自然数上的加法和乘法运算。)利用集合论公理(如 ZF 系统)可以证明 $\mathscr{N}$ 确是 PA 系统的模型(我们称之为**标准模型**),也就是说每

个 **PA** 定理都在 $\mathscr{N}$ 中为真。从而 **PA** 系统在语义上是可靠的，在语法上是一致的。于是，接下来的问题自然就是：

1. 在 $\mathscr{N}$ 中为真的命题是否都是 **PA** 定理？

2. 能不能只用 **PA** 公理而不用 **ZF** 公理证明 $\mathscr{N}$ 是 **PA** 模型？或者再稍弱一点，只用 **PA** 公理证明 **PA** 一致，即 **PA** 有模型（不管是不是 $\mathscr{N}$）？

哥德尔的两个不完全性定理就是分别对上述两个问题作出了否定的回答。

§6.2 PA 系统的算术化

为了在数论系统内讨论有关 **PA** 系统的问题，我们仿照第四章的方法将 **PA** 系统算术化，将有关 **PA** 系统的问题转化为有关自然数的问题。

首先，我们先使每个 **PA** 符号与一个自然数相联，即对每个初始符号 ξ，指定一个自然数 $\#'(\xi)$ 称作 ξ 的号码或 ξ 的哥德尔数，具体对应方法为：

ξ	$\neg$	$\to$	$\forall$	$=$	$\mathbf{0}$	S	$+$	$\cdot$	(	)	v_1	v_2	…
$\#'(\xi)$	0	1	2	3	4	5	6	7	8	9	10	11	…

表6.2.1　初始符号的号码

这个方法显然是能行的——对任何初始符号 ξ 我们可以根据表6.2.1算出它的号码；任给一个自然数我们也可以根据表6.2.1算出它代表哪个初始符号。

定义6.2.1　项的号码（哥德尔数）

1. 对复杂度为0的项 x、x 的号码 $\#(x)$ 定义为

$$\#(x)=J_1(0,0,\#'(x),0);$$

2. 设 t 为复杂度为 n 的项，其号码已定义，则复杂度为 $n+1$

的项 $S(t)$的号码定义为

$$\#(S(t))=J_4(0,5,\#(t),n+1);$$

3. 设 s,t 为复杂度小于等于 n 的项，$\#(s)$和$\#(t)$均已定义，则复杂度为 $n+1$的项$(s+t)$和$(s\cdot t)$的号码分别定义为

$$\#(s+t)=J_4(\#(s),6,\#(t),n+1);$$

$$\#(s\cdot t)=J_4(\#(s),7,\#(t),n+1)。$$

定义6.2.2 公式的号码(哥德尔数)

1. 设 s,t 是项，原子公式$(s=t)$的号码定义为

$$\#(s=t)=J_4(\#(s),3,\#(t),0);$$

2. 设 α 是复杂度为 n 的公式，$\#(\alpha)$已定义，则复杂度为 $n+1$ 的公式$\neg\alpha$ 的号码定义为

$$\#(\neg\alpha)=J_4(0,0,\#(\alpha),n+1);$$

3. 设 α,β 为复杂度小于等于 n 的公式，$\#(\alpha)$和$\#(\beta)$均已定义，则复杂度为 $n+1$的项$(\alpha\to\beta)$的号码定义为

$$\#(\alpha\to\beta)=J_4(\#(\alpha),1,\#(\beta),n+1);$$

4. 设 α 是复杂度为 n 的公式，$\#(\alpha)$已定义，则复杂度为 $n+1$ 的公式$\forall x\alpha$ 的号码定义为

$$\#(\forall x\alpha)=J_4(2,\#'(x),\#(\alpha),n+1)。$$

利用丘奇论题不难证明：

定理6.2.1 自然数上的二元谓词

x 是复杂度为 n 的项的号码

x 是复杂度为 n 的公式的号码

都是递归的。

定理6.2.2 自然数上的性质(一元谓词)

x 是项的号码

x 是公式的号码

都是递归的。

证 由定理6.2.1使用有界存在量词即得。

定义6.2.3(公式序列的号码) 设 $\varphi_1,\varphi_2,\cdots,\varphi_n$ 是长度为 n 的

公式序列，记作 Ψ，其号码定义为

$$\#(\Psi)=J_{n+1}(\#(\varphi_1),\#(\varphi_2),\cdots,\#(\varphi_n),n)。$$

通过上述定义6.2.1、定义6.2.2和定义6.2.3，每个项、每个公式、每个公式序列都与唯一的一个自然数(号码)相对应，今后我们就将这些对象与其号码等同看待，我们可以说"项 x"、"公式 x"、"公式序列 x"、"证明 x"(x 是自然数)，其严格意义分别是："号码为 x 的项"、"号码为 x 的公式"、"号码为 x 的公式序列"和"号码为 x 的证明(满足定义6.1.1的公式序列)"。

根据以上三个定义，任给一个有穷的公式序列 $\varphi_1,\varphi_2,\cdots,\varphi_n$，我们可以实际算出它的号码；反之，任给一个自然数 x，我们又可以使用如下的能行过程来确定它是否某个公式序列的号码，如果是，还可以实际写出这个公式序列。具体过程如下：

先计算 $r(x)$(这是个递归函数)，当 $r(x)=n>0$时，x 如果是某个公式序列的号码，则该公式序列的长度为 n；如果 $r(x)=0$，则 x 不是任何公式序列的号码。

当 $r(x)=n>0$时，依次计算 $\pi_1^{(n+1)}(x)$，$\pi_2^{(n+1)}(x)$，$\cdots$，$\pi_n^{(n+1)}(x)$(它们都是递归函数)，如果它们都是公式的号码(定理6.2.1和定理6.2.2)，则可以根据定义6.2.2具体写出这些公式；如果其中有一个不是公式的号码，则 x 不是公式序列的号码。这样我们就已使用丘奇论题证明了

定理6.2.3　自然数上的性质(一元谓词)

　　x 是公式序列(的号码)

是递归的。

再根据一阶逻辑中证明和演绎的可判定性，我们又可以由丘奇论题得到

定理6.2.4　自然数上的性质(一元谓词)

　　y 是一个 **PA** 证明(的号码)

和自然数上的二元谓词

　　y 是公式 x 的证明(的号码)

都是递归的。

§6.3 数字可表示性

上一节我们讨论了如何将有关 **PA** 系统的命题(元语言命题)转化为一个数论命题。本节将要讨论数论命题与 **PA** 公式的关系，特别是那些与元语言命题相对应的数论命题与 **PA** 公式的关系。

定义6.3.1 设 $A(x_1,\cdots,x_n)$是自然数上的 n 元谓词，如果存在恰含 n 个自由变元的 **PA** 公式 $\alpha(v_1,\cdots,v_n)$，使得对任意的自然数 $x_1,\cdots,x_n$，

(1) 如果 $A(x_1,\cdots,x_n)$为真，则 $\mathbf{PA}\vdash\alpha(\overline{x_1},\cdots,\overline{x_n})$，

并且

(2) 如果 $A(x_1,\cdots,x_n)$为假，则 $\mathbf{PA}\vdash\neg\alpha(\overline{x_1},\cdots,\overline{x_n})$，

(其中 $\overline{x}$ 是 x 的数字，见§6.1项的定义)则称 n 元谓词 $A(x_1,\cdots,x_n)$是**数字可表示**的谓词，称 $\alpha(v_1,\cdots,v_n)$是数字表示 $A(x_1,\cdots,x_n)$的公式。

定义6.3.2 设 $f(x_1,\cdots,x_n)$是 n 元(自然数)全函数，如果存在恰含 $n+1$ 个自由变元的 **PA** 公式 $\alpha(v_1,\cdots,v_n,v_{n+1})$ 使得对任何自然数 $x_1,\cdots,x_n,y$

(3) 如果 $f(x_1,\cdots,x_n)=y$，则 $\mathbf{PA}\vdash\alpha(\overline{x_1},\cdots,\overline{x_n},\overline{y})$，

(4) $\mathbf{PA}\vdash\exists! v_{n+1}\alpha(\overline{x_1},\cdots,\overline{x_n},v_{n+1})$。[①]

则称全函数 $f(x_1,\cdots,x_n)$是**数字可表示**的，$\alpha(v_1,\cdots,v_n,v_{n+1})$称为数字表示 $f(x_1,\cdots,x_n)$的公式。

请注意，定义6.3.2之(4)只要求对任何自然数 $x_1,\cdots,x_n$，

$$\mathbf{PA}\vdash\exists! v_{n+1}\alpha(\overline{x_1},\cdots,\overline{x_n},v_{n+1}),$$

并不要求

---

① $\exists!$表示“存在唯一的…”

$\mathbf{PA} \vdash \exists ! v_{n+1} \alpha(v_1, \cdots, v_n, v_{n+1})$。

显然后者蕴涵前者，而前者并不蕴涵后者。

由定义6.3.1容易证明

引理6.3.1　对任意常数 k，及任意的 $i(1 \leqslant i \leqslant n)$，如果

$$A(x_1, \cdots, x_{i-1}, x_i, x_{i+1}, \cdots, x_n)$$

是数字可表示的 n 元谓词，则

$$A(x_1, \cdots, x_{i-1}, k, x_{i+1}, \cdots, x_n)$$

是数字可表示的 $n-1$ 元谓词。

引理6.3.2　如果全函数 $f(x_1, \cdots, x_n)$ 是（由公式 $\alpha(v_1, \cdots, v_n, v_{n+1})$）数字可表示的，则谓词 $f(x_1, \cdots, x_n) = y$ 也是由公式 $\alpha(v_1, \cdots, v_n, v_{n+1})$ 数字表示的。

证　由定义6.3.2只须证明

如果 $f(x_1, \cdots, x_n) \neq y$，则 $\mathbf{PA} \vdash \neg \alpha(\overline{x_1}, \cdots, \overline{x_n}, \overline{y})$。

设 $f(x_1, \cdots, x_n) \neq y$，由于 f 是全函数，于是有自然数 z 使 $f(x_1, \cdots, x_n) = z$ 且 $y \neq z$。而二元谓词 $y \neq z$ 是由公式

$$v_{n+1} \neq v_{n+2}$$

数字表示的（请读者自行验证），于是有

$$\mathbf{PA} \vdash \overline{y} \neq \overline{z}$$

再由定义6.3.2有

$$\mathbf{PA} \vdash \alpha(\overline{x_1}, \cdots, \overline{x_n}, \overline{z}),$$

及

$$\mathbf{PA} \vdash \exists ! v_{n+1} \alpha(\overline{x_1}, \cdots, \overline{x_n}, v_{n+1}),$$

由以上三式利用谓词演算容易得到

$$\mathbf{PA} \vdash \neg \alpha(\overline{x_1}, \cdots, \overline{x_n}, \overline{y})。$$

本节的主定理是：

定理6.3.3　每个递归谓词都是数字可表示的。

由于一个谓词 A 的递归性与其特征函数 C_A 的递归性是一回事，根据引理6.3.1和6.3.2我们只须证明每个递归函数都是数字可表示的就可以了。我们按照递归函数定义的四个部分来证明本

定理，即先证明初始函数都是数字可表示的，再证明数字可表示函数类对复合、原始递归和取极小运算封闭。为了便于阅读，我们把定理4.3的证明分成若干引理来陈述。

引理6.3.4 初始函数都是数字可表示的。

证 1. 零函数 $O(x)=0$ 是数字可表示的，因为公式

$$v_2=0$$

是数字表示一元函数 $O(x)$ 的 PA 公式：

当 $y=0$ 时，PA $\vdash \mathbf{0}=\mathbf{0}$ （公理 $Ax7$）；

我们当然也有

对任何自然数 x，PA $\vdash \exists! v_2(v_2=\mathbf{0})$。

2. 后继函数 $S(x)=x+1$ 是数字可表示的，因为公式

$$v_2=S(v_1)$$

是数字表示一元函数 $x+1$ 的 PA 公式：

当 $y=x+1$ 时，PA $\vdash \overline{y}=S(\overline{x})$ （公理 $Ax7$）；

我们当然也可以证明对任何自然数 x

$$\text{PA} \vdash \exists! v_2(v_2=S(\overline{x}))。$$

3. 每个投影函数 $U_j^n(x_1,\cdots,x_n)=x_j$ 是数字可表示的，公式

$$v_{n+1}=v_j$$

是表示 $n+1$ 元关系 $U_j^n(x_1,\cdots,x_n)=y$ 的 PA 公式：

当 $y=x_j$ 时，PA $\vdash \overline{y}=\overline{x_j}$；

对任何自然数 $x_1,\cdots,x_n$，当然有

$$\text{PA} \vdash \exists! v_{n+1}(v_{n+1}=\overline{x_j})。$$

引理6.3.5 数字可表示函数对复合运算封闭。

证 设 $f(y_1,\cdots,y_m)$，$g_1(x_1,\cdots,x_n)$，$\cdots$，$g_m(x_1,\cdots,x_n)$ 都是数字可表示函数，公式 $\alpha(v_1,\cdots,v_m,v_{m+1})$，$\beta_1(v_1,\cdots,v_n,v_{n+1})$，$\cdots$，$\beta_m(v_1,\cdots,v_n,v_{n+1})$ 分别是数字表示函数 $f(y_1,\cdots,y_m)$，$g_1(x_1,\cdots,x_n)$，$\cdots$，$g_m(x_1,\cdots,x_n)$ 的 PA 公式，则容易验证 PA 公式

$$\beta_1(v_1,\cdots,v_{n+1}) \wedge \beta_2(v_1,\cdots,v_{n+2}) \wedge \cdots \wedge \beta_m(v_1,\cdots,v_{n+m})$$

$\wedge \alpha(v_{n+1},\cdots,v_{n+m+1})$

是数字表示 n 元函数

$$f(g_1(x_1,\cdots,x_n),\cdots,g_m(x_1,\cdots,x_n))$$

的公式。

原始递归运算的情况比较复杂,我们先要做一些准备工作。

引理6.3.6　二元函数 $x+y$, $x\cdot y$ 都是数字可表示的。

证　函数 $x+y$ 由 **PA** 公式 $v_1+v_2=v_3$数字表示;函数 $x\cdot y$ 则由 **PA** 公式 $v_1\cdot v_2=v_3$数字表示(请读者自行验证)。

考虑自然数上的二元全函数

$$\text{rem}(x,y)=\begin{cases} x \text{ 被 } y \text{ 整除时的余数} & \text{当 } y\neq 0\text{时},\\ x & \text{当 } y=0\text{时}。\end{cases}$$

引理6.3.7　二元函数 $\text{rem}(x,y)$是数字可表示的。

证　**PA** 公式

$$\exists v_4(v_3<v_2\wedge v_1=v_2\cdot v_4+v_3)\vee(v_2=0\wedge v_3=v_1)$$

是数字表示自然数函数 $\text{rem}(x,y)$的。其中"$v_3<v_2$"是公式

$$\exists v_5(v_5\neq 0\wedge v_3+v_5=v_2)$$

的缩写。

考虑三元函数 $\beta(x,y,z)$:

$$\beta(x,y,z)=\text{rem}(x,(y+1)\cdot z+1)$$

由引理6.3.5引理6.3.6和引理6.3.7可知 $\beta(x,y,z)$是数字可表示的。这个函数称为哥德尔 β 函数。为方便起见我们将数字表示 $\beta(x,y,z)$的 **PA** 公式就记作 $\beta^*(v_1,v_2,v_3,v_4)$。

在这里我们要用到初等数论中的一个著名定理

引理6.3.8(**孙子定理、中国余数定理**)　令 $a_1,a_2,\cdots,a_n$ 是任意的 n 个自然数, $m_1,m_2,\cdots,m_n$ 是两两互素的 n 个自然数,则存在一个自然数 x 使得对任何 $i=1,2,\cdots,n$ 有

$$x\equiv a_i(\text{mod } m_i)^{①}。$$

① $x\equiv y(\text{mod } m)$的意思是:x,y 分别被 m 除时,余数相等,即 m 可以整除 $x-y$。

证 对 n 作归纳。

当 $n=1$时，取 $x=a_1$，当然有 $x\equiv a_1(\mathrm{mod}\ m_1)$。

设当 $n=k$ 时结论已证，即已有 x_0满足

$$x_0\equiv a_1(\mathrm{mod}\ m_1),$$

$$x_0\equiv a_2(\mathrm{mod}\ m_2),$$

$$x_0\equiv a_3(\mathrm{mod}\ m_3),$$

…………

$$x_0\equiv a_k(\mathrm{mod}\ m_k)。$$

我们来构造 x，使 x 不仅满足上面 k 个同余式，而且满足

$$x\equiv a_{k+1}(\mathrm{mod}\ m_{k+1})。$$

由于 $m_1,\cdots,m_{k+1}$两两互素，所以 $m_1m_2\cdots m_k$ 与 m_{k+1}互素，于是有整数(正的、负的、零)r 和 s 使

$$rm_1\cdots m_k+sm_{k+1}=1。$$

从而

$$r(a_{k+1}-x_0)m_1\cdots m_k+s(a_{k+1}-x_0)m_{k+1}=a_{k+1}-x_0,$$

记 $q=r(a_{k+1}-x_0)$，$p=-s(a_{k+1}-x_0)$，则有

$$x_0+qm_1\cdots m_k=a_{k+1}+pm_{k+1},$$

适当选择 t，使 $tm_{k+1}+q\geqslant 0$，则

$$x_0+(tm_{k+1}+q)m_1\cdots m_k=a_{k+1}+m_{k+1}(p+tm_1\cdots m_k)\ 。$$

取

$$x=x_0+(tm_{k+1}+q)m_1\cdots m_k$$

即可。

引理6.3.9 任给自然数 $a_0,a_1,\cdots,a_n$，存在自然数 x,y，使对每个 $i(i=0,1,2,\cdots,n)$有

$$\beta(x,y,i)=a_i。$$

证 令 $s=\max\{n,a_0,a_1,\cdots,a_n\}$ 。

考虑

$$m_i=(i+1)s!+1\quad(i=0,1,2,\cdots,n)。$$

这 $n+1$个数是两两互素的。因为如果 m_i 和 m_{i+k}有素的公因数 p,

即

$$p|(i+1)s!+1 \quad 且 \quad p|(i+k+1)s!+1,$$

则 $p|k\cdot s!$。由于 $s>k$,故有 $p|s!$。这与 $p|(i+1)s!+1$ 及 p 是素数矛盾。

利用引理6.3.8,有自然数 x 使

$$x\equiv a_0(\mathrm{mod}\ m_0),$$

$$x\equiv a_1(\mathrm{mod}\ m_1),$$

$$\cdots\cdots\cdots\cdots$$

$$x\equiv a_n(\mathrm{mod}\ m_n)。$$

由于 $a_i<m_i$, 令 $y=s!$ 于是对每个 $i(i=0,1,\cdots,n)$有

$$\begin{aligned}\beta(x,y,i)&=\mathrm{rem}(x,(i+1)y+1)\\&=\mathrm{rem}(a_i,(i+1)y+1)\\&=a_i。\end{aligned}$$

引理6.3.10 数字可表示函数对原始递归运算封闭。

证 设 $g(x_1,\cdots,x_n)$, $h(x_1,\cdots,x_n,y,z)$都是数字可表示的函数,

$$\theta(v_1,\cdots,v_n,v_{n+1}) \quad 和 \quad \eta(v_1,\cdots,v_n,v_{n+1},v_{n+2},v_{n+3})$$

分别是数字表示它们的**PA**公式, $f(x_1,\cdots,x_n,y)$是由 g 和 h 经原始递归而得到的 $n+1$元函数。

任给 $x_1,\cdots,x_n,y$ 我们有 $y+1$个自然数 $a_0,a_1,\cdots,a_y$,使

$$a_0=f(x_1,\cdots,x_n,0)=g(x_1,\cdots,x_n),$$

$$a_1=f(x_1,\cdots,x_n,1)=h(x_1,\cdots,x_n,0,a_0),$$

$$\cdots\cdots\cdots\cdots$$

$$a_y=f(x_1,\cdots,x_n,y)=h(x_1,\cdots,x_n,y-1,a_{y-1})。$$

上述事实可以写作

$$\exists c\exists d\{\beta(c,d,0)=g(x_1,\cdots,x_n)\wedge\forall i[i<y \rightarrow\beta(c,d,i+1)=h(x_1,\cdots,x_n,i,\beta(c,d,i))]\}$$

于是**PA**公式

$$\exists v_{n+3}\exists v_{n+4}\{\exists v_{n+5}[\beta^*(v_{n+3},v_{n+4},\mathbf{0},v_{n+5})\wedge\theta(v_1,\cdots,$$

$v_n, v_{n+5})] \wedge \forall v_{n+7}[v_{n+7} < v_{n+1} \rightarrow \exists v_{n+5} \exists v_{n+6}[\beta^*(v_{n+3}, v_{n+4}, i+1, v_{n+5}) \wedge \beta^*(v_{n+3}, v_{n+4}, i, v_{n+6}) \wedge \eta(v_1, \cdots, v_n, i, v_{n+6}, v_{n+5})]] \wedge \beta^*(v_{n+3}, v_{n+4}, v_{n+1}, v_{n+2})\}$

是数字表示 $f(x_1, \cdots, x_n, y)$的(请读者自行验证)。

引理6.3.11　如果 $g(x_1, \cdots, x_n, y)$是数字可表示的函数，

$$f(x_1, \cdots, x_n) = \mu y[g(x_1, \cdots, x_n, y) = 0]$$

仍是全函数,则$f(x_1, \cdots, x_n)$ 也是数字可表示的。

证　设 $\alpha(v_1, \cdots, v_n, v_{n+1}, v_{n+2})$是数字表示 $n+1$元函数 g 的 **PA** 公式,则公式

$\alpha(v_1, \cdots, v_n, v_{n+1}, 0) \wedge \forall v_{n+3}[v_{n+3} < v_{n+1} \rightarrow \neg\alpha(v_1, \cdots, v_n, v_{n+3}, 0)]$

是数字表示 f 的 **PA** 公式。

至此我们已经证明每个递归全函数都是数字可表示的。因而每个递归谓词也都是数字可表示的。这就完成了定理6.3.3的证明。

§6.4　哥德尔不完全性定理

有了以上几节的准备,我们现在就可以讨论哥德尔不完全性定理了。

定义6.4.1　设 $\alpha(v)$为恰含一个自由变元 v 的 **PA** 公式，$x = \#(\alpha)$,将 x 的数字 $\overline{x}$ 代入 $\alpha(v)$所得到的公式 $\alpha(v/\overline{x})$称作 α 的**自代入**。

一个恰含一个自由变元的公式的自代入仍是一个公式,不过原来的自由变元已被无变元项 $\overline{x}$ 所取代,成为比原公式长得多的一个闭公式。

我们来考虑自然数上的二元谓词 $A(x,y)$:

y 是恰含一个自由变元的公式 x 的自代入的证明。

根据§6.3中的约定,我们将一个公式与其号码等同看待,说

“公式 x ”就是说“号码为 x 的公式”；说“证明 y”就是说“号码为 y 的证明”。因此二元关系 $A(x,y)$ 的完整含义就是：

> x 是某个恰含一个自由变元的公式 α 的号码并且 y 是 α 的自代入的证明的号码。

容易看出，二元关系 $A(x,y)$是能行可判定的。例如可以用以下方法来判定 $A(x,y)$的真假：

1. 先确定 x 是否某个公式的号码，由定理6.2.2，这是递归的，因而是能行的。如果不是，则 $A(x,y)$为假；如果是，则继续

2. 具体写出号码为 x 的公式，由定义6.2.2，这个过程是能行的，看该公式（为说话方便，记此公式为 α）是否恰含一个自由变元。如果不是，则 $A(x,y)$为假；如果是，则继续

3. 将 x 的数字 $\overline{x}$ 代入 α，求得其自代入 $\alpha(v/\overline{x})$，然后计算其号码 $\#(\alpha(v/\overline{x}))$，这个过程当然是能行。再继续

4. 确定 y 是否 $\#(\alpha(v/\overline{x}))$的证明的号码，由定理6.2.4，这是递归的因而是能行的。如果不是，则 $A(x,y)$ 为假；如果是，则 $A(x,y)$为真。

既然二元谓词 $A(x,y)$是能行可判定的，根据丘奇论题，谓词 $A(x,y)$是递归的，因此由定理6.3.3知 $A(x,y)$是数字可表示的。设恰含两个自由变元的 **PA** 公式 $\gamma(v_1,v_2)$是数字表示 $A(x,y)$的公式，根据定义6.3.1我们有

(5.1)　如果 $A(x,y)$真（即“y 是 x 的自代入的证明”），则

$$\mathbf{PA}\vdash\gamma(\overline{x},\overline{y}),$$

并且

(5.2)　如果 $A(x,y)$假（即“y 不是 x 的自代入的证明”），则

$$\mathbf{PA}\vdash\neg\gamma(\overline{x},\overline{y})。$$

接下来，我们再考虑恰含一个自由变元的公式

$$\forall v_2\neg\gamma(v_1,v_2)。$$

为了说话方便，将此公式记作 $\alpha(v_1)$。根据上面所说的数字可表示性，此公式可以读作“任何公式序列 v_2都不是公式 v_1的自代入

的证明”,这等价于说“公式 v_1的自代入是不可证的”。

公式 $\alpha(v_1)$当然也有一个号码,记作 p,即

$$p = \# \alpha(v_1),$$

考虑 α 的自代入

$$\alpha(v_1/\overline{p}), \qquad (*)$$

它的直观意义可以理解为“号码为 p 的公式的自代入是不可证的”,而“号码为 p 的公式的自代入”就是 $\alpha(v_1/\overline{p})$自身,所以公式(*)的直观意义就是说它自己不可证。这就导致了哥德尔不完全性定理的一个方面:

定理6.4.1 如果 **PA** 是一致的,则句子(闭公式)$\alpha(v_1/\overline{p})$在 **PA** 中不可证。

证 用反证。设 $\alpha(v_1/\overline{p})$在 **PA** 中可证,即

$$\mathbf{PA} \vdash \alpha(v_1/\overline{p}),$$

亦即

$$\mathbf{PA} \vdash \forall v_2 \neg \gamma(\overline{p}, v_2)。$$

由逻辑公理 $Ax4$知,对每个无变元项 $\overline{y}$,都有

$$\mathbf{PA} \vdash \neg \gamma(\overline{p}, \overline{y})。$$

由于 **PA** 一致,从而对每个无变元项 $\overline{y}$,都有

$$\mathbf{PA} \nvdash \gamma(\overline{p}, \overline{y})。$$

而 γ 是数字表示自然数二元关系 A 的,于是由(5.1)就有:

对每个自然数 y, $A(p,y)$不为真,

即

对每个自然数 y, y 都不是 p 的自代入的证明,

也就是说

p 的自代入不可证。

但我们已经知道 p 的自代入就是 $\alpha(v_1/\overline{p})$,所以 $\alpha(v_1/\overline{p})$不可证。这与开始时所假设的 $\alpha(v_1/\overline{p})$在 **PA** 中可证矛盾。

接下来的工作是要证明句子$\neg\alpha(v_1/\overline{p})$在 **PA** 中也不可证。为此先要引人 ω-一致性的概念。

定义6.4.2　如果对任何恰含一个自由变元的PA公式$\theta(v_1)$，句子

$$\theta(\overline{0}),\ \theta(\overline{1}),\ \theta(\overline{2}),\cdots,\theta(\overline{n}),\cdots \text{和} \neg\forall v_1\theta(v_1)$$

不都是PA可证的，则称PA系统是**ω-一致的**；否则称PA系统是ω-不一致的。

容易看出ω-一致性是比一致性更强的要求：如果PA是ω-一致的，则它也是一致的。哥德尔不完全性定理的另一个方面是：

定理6.4.2　如果PA系统是ω-一致的，则句子$\neg\alpha(v_1/\overline{p})$在PA中不可证。

证　仍用反证。设$\neg\alpha(v_1/\overline{p})$在PA中可证，即

$$\mathbf{PA}\vdash\neg\alpha(v_1/\overline{p}) \qquad (**)$$

亦即

$$\mathbf{PA}\vdash\neg\forall v_2\neg\gamma(\overline{p},v_2),$$

则由PA的ω-一致性知句子

$$\neg\gamma(\overline{p},\overline{0}),\ \neg\gamma(\overline{p},\overline{1}),\cdots,\ \neg\gamma(\overline{p},\overline{n}),\cdots$$

中至少有一个在PA中不可证。不妨设$\neg\gamma(\overline{p},\overline{k})$在PA中不可证。则由(5.2)知$A(p,k)$不为假，即$A(p,k)$为真。这就是说

k是p的自代入的证明

为真，从而p的自代入在PA中可证。而我们已经知道就p的自代入就是$\alpha(v_1/\overline{p})$，于是

$$\mathbf{PA}\vdash\alpha(v_1/\overline{p})。 \qquad (***)$$

(**)、(***)两式与PA系统的一致性(ω-一致性)矛盾。

定理6.4.2和定理6.4.3合称哥德尔不完全性定理，在一般文献中是如下陈述的：

定理6.4.3(**哥德尔不完全性定理**)　存在一个PA句子ξ，使得：如果PA是一致的，则ξ在PA中不可证；如果PA是ω-一致的，则$\neg\xi$在PA中不可证。

因此PA是不完全的。

这种在一个形式系统中既不可证明又不可否证的句子称为在

该系统中形式不可判定语句，也叫哥德尔语句。

哥德尔在证明定理6.4.3的后一半(即定理6.4.2)时使用了比一致性更强的 ω-一致性作为条件。后来，罗塞证明 ω-一致性的条件并不必要，他另构造了一个(更为复杂的)哥德尔语句 δ，在 **PA** 一致的条件下证明了 δ 和 $\neg\delta$ 都不可证。罗塞的 δ 是如下得到的：

设 $A(x,y)$ 仍是前面所说的自然数上的二元关系：

> y 是 **PA** 公式 x 的自代入的证明，

$B(x,y)$ 是下述二元关系：

> y 是公式 x 的自代入的否定式的证明。

容易看出 $A(x,y)$ 和 $B(x,y)$ 都是递归的，因而都是数字可表示的，设恰含两个自由变元的 **PA** 公式 $\gamma(v_1,v_2)$ 和 $\gamma'(v_1,v_2)$ 分别数字表示 $A(x,y)$ 和 $B(x,y)$，于是我们有

(5.3)　如果 $A(x,y)$ 真，则 $\mathbf{PA}\vdash\gamma(x,y)$；

(5.4)　如果 $A(x,y)$ 假，则 $\mathbf{PA}\vdash\neg\gamma(x,y)$；

(5.5)　如果 $B(x,y)$ 真，则 $\mathbf{PA}\vdash\gamma'(x,y)$；

(5.6)　如果 $B(x,y)$ 假，则 $\mathbf{PA}\vdash\neg\gamma'(x,y)$。

考虑 **PA** 公式

$$\forall v_2[\gamma(v_1,v_2)\rightarrow\exists v_3(v_3\leqslant v_2\wedge\gamma'(v_1,v_3))]$$

此公式的号码记作 q，q 的自代入即为句子 δ：

$$\forall v_2[\gamma(\bar{q},v_2)\rightarrow\exists v_3(v_3\leqslant v_2\wedge\gamma'(\bar{q},v_3))]。$$

δ 的意思可以看作是：

> 如果 q 的自代入有一个证明，则其否定式有一个号码更小的证明。

现在，我们可以证明

定理6.4.5　不完全性定理(罗塞，1936)存在一个 **PA** 句子 δ 使得如果 **PA** 一致，则 δ 和 $\neg\delta$ 在 **PA** 中都不可证。因此 **PA** 是不完全的。

证　我们来证明对上述 δ，δ 和 $\neg\delta$ 都是在 **PA** 中不可证的。仍用反证。

假如 δ 可证，即

$$\mathbf{PA}\vdash\forall v_2[\gamma(\overline{q},v_2)\to\exists v_3(v_3\leqslant v_2\wedge\gamma'(\overline{q},v_3))]。\qquad ①$$

设 δ 的一个证明的号码为 k。由于 δ 是 q 的自代入，所以 $A(q,k)$ 为真。由(5.3)可得

$$\mathbf{PA}\vdash\gamma(\overline{q},\overline{k})。\qquad ②$$

另外，由逻辑公理 $Ax4$和①，对无变元项 $\overline{k}$ 有

$$\mathbf{PA}\vdash\gamma(\overline{q},\overline{k})\to\exists v_3(v_3\leqslant\overline{k}\wedge\gamma'(\overline{q},v_3))\qquad ③$$

于是由②，③可得

$$\mathbf{PA}\vdash\exists v_3(v_3\leqslant\overline{k}\wedge\gamma'(\overline{q},v_3))$$

由于在 PA 中可证

$$\mathbf{PA}\vdash v_3\leqslant\overline{k}\to v_3=\overline{0}\vee v_3=\overline{1}\vee v_3=\overline{2}\vee\cdots\vee v_3=\overline{k}$$

因此可以得到

$$\mathbf{PA}\vdash\gamma'(\overline{q},\overline{0})\vee\gamma'(\overline{q},\overline{1})\vee\cdots\vee\gamma'(\overline{q},\overline{k})$$

也就是

$$\mathbf{PA}\vdash\neg(\neg\gamma'(\overline{q},\overline{0})\wedge\neg\gamma'(\overline{q},\overline{1})\wedge\cdots\wedge\neg\gamma'(\overline{q},\overline{k}))。$$

由于 **PA** 一致，从而对某个自然数 $r\leqslant k$ 有

$$\mathbf{PA}\nvdash\neg\gamma'(\overline{q},\overline{r})。$$

于是由(5.6)知 $B(q,r)$不为假，即 $B(q,r)$为真，也就是说 q 的自代入的否定(即$\neg\delta$)可证。这与 **PA** 的一致性矛盾。

假如$\neg\delta$ 可证，即

$$\mathbf{PA}\vdash\neg\forall v_2(\gamma(\overline{q},v_2)\to\exists v_3(v_3\leqslant v_2\wedge\gamma'(\overline{q},v_3)))$$

也就是

$$\mathbf{PA}\vdash\exists v_2(\gamma(\overline{q},v_2)\wedge\forall v_3(\gamma'(\overline{q},v_3)\to v_3>v_2))\qquad ④$$

设$\neg\delta$ 的一个证明的号码为 m，由于$\neg\delta$ 是 q 的自代入的否定，所以 $B(q,m)$为真，由(5.5)可得

$$\mathbf{PA}\vdash\gamma'(\overline{q},\overline{m})\qquad ⑤$$

根据谓词演算，由④，⑤可得

$$\mathbf{PA}\vdash\exists v_2(\gamma(\overline{q},v_2)\wedge(\gamma'(\overline{q},\overline{m})\to\overline{m}>v_2))\qquad ⑥$$

再经过一系列相当烦琐的谓词演算过程(读者可自己试一下)可得

$$\mathbf{PA} \vdash \gamma(\bar{q},\bar{0}) \lor \gamma(\bar{q},\bar{1}) \lor \cdots \lor \gamma(\bar{q},\overline{m-1})$$

也就是

$$\mathbf{PA} \vdash \neg(\neg\gamma(\bar{q},\bar{0}) \land \neg\gamma(\bar{q},\bar{1}) \land \cdots \land \neg\gamma(\bar{q},\overline{m-1}))。$$

再根据 **PA** 的一致性知对某个自然数 $n<m$ 有

$$\mathbf{PA} \nvdash \neg\gamma(\bar{q},\bar{n})。$$

从而由(5.4)知 $A(q,n)$不为假，即 $A(q,n)$为真，也就是说 q 的自代入 δ 可证，与 **PA** 的一致性矛盾。

事实上，哥德尔不完全性定理的意义并不局限于 **PA** 系统不完全这一结论。哥德尔的证明方法具有普遍意义，对任何一个强于 **PA** 的形式系统(比如 **ZF** 系统)，我们都可以利用上述方法构造出一个哥德尔语句 α，使得 α 和$\neg\alpha$ 在该系统中都不可证。因此哥德尔不完全性定理又可以陈述为

定理6.4.5　对任何一个足够强(即包含形式数论作为其子系统)的形式理论 **S**，如果 **S** 是一致的，则 **S** 不完全。

在证明以上各定理的过程中，我们构造了一些在 **PA** 中形式不可判定的语句。这些语句的逻辑意义十分明显，但却没有数学意义。它们都是为了证明 **PA** 的不完全性而刻意制造出来的。对于研究数论的人，无论他是否采用形式化的方法，都不会遇到这些命题。所以自从哥德尔不完全性定理发表以来，人们一直想知道是否有一个自然的、在数论研究中有意义的自然数命题是在 **PA** 系统中形式不可判定的?1970年马蒂亚谢维奇在 **M**·戴维斯、**J**·罗宾逊和 **H**·普特南的工作的基础之上证明“每个递归可枚举谓词都是刁番图谓词”(从而最终否定地解决了希尔伯特第十问题)。这项结论的一个推论是：

*存在一个整系数多项式方程 $D=0$，它没有正整数解，但在 **PA** 中却不能证明这一事实。*

这表明 $D\neq0$ 的全称闭包是一个哥德尔语句。这是一个地道的数学命题，而且 $D\neq0$ 可以实际地用显式写出来。但是这个公式是个庞然大物——有数百个字符，并且它本身也是根据哥德尔(第

二)不完全性定理翻造出来的，因而不能算作自然的数学命题。直到1977年，J·帕瑞斯和L·哈灵顿才在组合论中找到了一个命题α，α在**ZF**中可证但在**PA**中不可证。这个组合论命题α可以认为是一个自然的哥德尔语句。由于本书并不假设读者具备组合论的预备知识，所以就不具体介绍α的构造了。

上面所讨论的哥德尔不完全性定理(定理6.4.3至定理6.4.5)也称作**哥德尔第一不完全性定理**。它说

如果"**PA**一致"则"p的自代入不可证"(定理6.4.2)

我们对定理6.4.4的证明是在元语言的层次上进行的。我们知道"**PA**一致"和"p的自代入不可证"都可以在**PA**中表达，后者的表达式就是$\alpha(v_1/\overline{p})$，前者的表达式我们记作

Consis(**PA**)

(读者可参照定理6.4.4的证明自己试着写一下)。于是定理6.4.2可以表达作

Consis(**PA**)$\to\alpha(v_1/\overline{p})$。

如果我们能在**PA**内部完成对上式的证明，即得到

PA$\vdash$Consis(**PA**)$\to\alpha(v_1/\overline{p})$，

那么由于我们已经证明：如果**PA**一致，则**PA**$\vdash\alpha(v_1/\overline{p})$。于是就得到：

定理6.4.6　如果**PA**是一致的，那么**PA**的一致性不能在**PA**内部证明。

这是一条意义更为深远的定理，通常称为**哥德尔第二不完全性定理**。哥德尔并没有给出这个定理的完整证明，只是象上面那样大体上勾勒出了证明的思路。他原想另写一篇文章来证明这一定理。但他的结果发表以后立即得到了广泛的承认，于是这计划中的第二篇文章也就一直没有完成。此后，希尔伯特和贝奈斯(1939年)、菲弗尔曼(1960年)都曾给出定理6.4.6的严格证明，每个证明均长达四十余页。

第七章

递归可枚举集

§7.1 判定问题

根据丘奇论题，说一个谓词(集合)是可判定的，就是说它是递归的。例如前面我们已经证明了谓词“x是偶数”、“x是素数”、“x能整除y”、“$x<y$”都是递归谓词，因而它们都是可判定的。当然，并非所有的数论谓词(集合)都是可判定的。下面我们先来看几个与递归论有关的不可判定问题。

定理7.1.1　谓词“$x\in W_x$”(即“$\Phi_x(x)\downarrow$”)不是递归的。

证　用反证。设“$x\in W_x$”是递归集，则其特征函数

$$f(x)=\begin{cases}1 & \text{如果 } x\in W_x\\ 0 & \text{如果 } x\notin W_x\end{cases}$$

是递归函数。定义函数

$$g(x)=\begin{cases}0 & \text{如果 } f(x)=0,\\ \text{无定义} & \text{否则。}\end{cases}$$

根据第三章习题12，g是递归函数，于是对某个自然数m，$g=\Phi_m$ ($\text{dom } g=W_m$)。此时

$$m\in W_m\Leftrightarrow m\in\text{dom } g\Leftrightarrow f(m)=0\Leftrightarrow m\notin W_m,$$

矛盾。这就表明$x\in W_x$不是递归谓词。

定理7.1.2(**停机问题**不可判定)　谓词“$y\in W_x$”(即$\Phi_x(y)\downarrow$)不是递归的。

证　用反证。设$y\in W_x$是递归谓词，则其特征函数

$$g(x,y)=\begin{cases}1 & \text{如果 } y\in W_x, \\ 0 & \text{如果 } y\notin W_x。\end{cases}$$

是递归函数。于是一元递归函数

$$f(x)=g(x,x)$$

也是递归函数，而 f 正是谓词 $x\in W_x$ 的特征函数，上一定理已经证明它不是递归的。这个矛盾就表明 $y\in W_x$ 不是递归谓词。

定理7.1.3　谓词"$\Phi_x=O$(零函数)"不是递归的。

证　令

$$f(x,y)=\begin{cases}0 & \text{如果 } x\in W_x, \\ \text{无定义} & \text{如果 } x\notin W_x,\end{cases}$$

根据丘奇论题，$f(x,y)$是递归函数。由 s-m-n 定理，存在递归全函数 $t(x)$，使

$$f(x,y)=\Phi_{t(x)}(y)。\qquad ①$$

由于当 $x\in W_x$ 时，$f(x,y)$作为 y 的函数是零函数，从而由①知

$$x\in W_x\Leftrightarrow \Phi_{t(x)}=O。$$

而如果 $\Phi_x=O$ 是递归的，$\Phi_{t(x)}$当然也是递归的，于是 $x\in W_x$ 是递归的，与定理7.1.1矛盾，所以 $\Phi_x=O$ 不是递归谓词。

推论7.1.4　谓词"$\Phi_x=\Phi_y$"不是递归的。

证　令 e 是零函数的一个指标，如果 $\Phi_x=\Phi_y$ 递归，则 $\Phi_x=\Phi_e$ 也是递归的，与定理7.1.3矛盾。

定理7.1.5(**输出问题**不可判定)　设 c 为常数，谓词$\exists y(\Phi_x(y)\downarrow=c)$(即"第 x 号图灵机可以输出 c"亦即 $c\in E_x$)不是递归的。

证　令

$$f(x,y)=\begin{cases}y & \text{如果 } x\in W_x, \\ \text{无定义} & \text{否则。}\end{cases}$$

显然 $f(x,y)$是二元递归函数。于是由 s-m-n 定理，存在递归全函数 $t(x)$使

$$f(x,y)=\Phi_{t(x)}(y)。$$

从而

$$x\in W_x \Rightarrow \Phi_{t(x)}(y)=y \Rightarrow E_x=N \Rightarrow c\in E_{t(x)},$$

$$x\notin W_x \Rightarrow \Phi_{t(x)}(y)\text{处处无定义} \Rightarrow E_x=\varnothing \Rightarrow c\notin E_{t(x)}。$$

因此

$$x\in W_x \Leftrightarrow c\in E_{t(x)}。$$

如果 $c\in E_x$ 是递归的，则 $c\in E_{t(x)}$ 也是递归的，从而 $x\in W_x$ 也是递归的，这与定理7.1.1矛盾。

定理7.1.6(**输入问题**不可判定) 设 c 为常数，谓词 $\Phi_x(c)\downarrow$ (即 $c\in W_x$)不是递归的。

证明留给读者(习题1)。

上述六个定理表明在递归论中存在大量的不可判定的问题。通过编号的方法，我们也可以证明许多关于形式系统、形式理论的问题也是不可判定的(当然，也有许多是可判定的)。由于许多证明都要涉及递归论以外的知识，而且篇幅也很大，所以我们只是不加证明地列出一些重要的结果(参看[4],[1])。

对于一个形式系统或形式理论，用能行的方法确定任意的公式是否该系统的定理，称为该系统的**判定问题**。如果有这样的能行方法存在(即该系统的定理集是递归的)，就称该系统是可判定的，否则称为不可判定的。我们有如下的一些结果：

1. 经典命题逻辑是可判定的(例如真值表法就是一种判定方法)；

2. 直觉主义命题逻辑是可判定的；

3. 模态逻辑的 K，T，S_2，S_3，S_4，S_5等系统都是可判定的；

4. 一阶谓词逻辑是不可判定的。

不过如果把谓词逻辑的判定问题限制在某些特殊的公式类上，仍有一些可判定的结果，例如

5. 对只含一元谓词的公式，其有效性和可满足性都是可判定的；

7. 对形如$\forall x_1\cdots\forall x_n\alpha$ 和形如$\exists x_1\cdots\exists x_n\alpha$($\alpha$ 中无量词)的谓词

公式，其有效性和可满足性都是可判定的；

7. 对形如$\forall x_1\cdots\forall x_n\exists y_1\cdots\exists y_m\alpha$($\alpha$中无量词)的谓词公式，其有效性是可判定的，但其可满足性不是可判定的；

8. 对形如$\exists x_1\cdots\exists x_n\forall y_1\cdots\forall y_m\alpha$($\alpha$中无量词)的谓词公式，其可满足性是可判定的，但其有效性不是可判定的。

对于建立在一阶逻辑基础上的各种形式理论，其判定问题因其非逻辑公理的不同而各异。例如

9. 一阶数论是不可判定的；

10. 群论是不可判定的；

11. 交换群论是可判定的；

12. 实闭有序域理论(实数算术)是可判定的。

§7.2 递归可枚举集

上一节中我们讨论了一些不可判定(非递归)的谓词，如果细心观察一下就会发现它们之中仍有很大的差别。我们来对比一下"$\Phi_x(x)\downarrow$"，"Φ_x处处无定义"和"$\Phi_x=\Phi_y$"三个谓词，它们都是非递归的(不可判定的)。

对"$\Phi_x(x)\downarrow$"，在给定x之后，我们可以实际写出图灵机Z使$\#(Z)=x$，然后输入x，运行Z，这是一个能行的过程。如果事实上$\Phi_x(x)\downarrow$，那么使用上述过程我们可以在有穷步内确认$\Phi_x(x)\downarrow$的事实，因为此时Z会停机；如果事实上$\Phi_x(x)\uparrow$，则Z不能停机，我们在有穷步内无法作出任何结论。也就是说，虽然谓词"$\Phi_x(x)\downarrow$"不是可判定，但对"$\Phi_x(x)\downarrow$"为真的这一半我们还是可以通过一个能行过程加以肯定的，只是对"$\Phi_x(x)\downarrow$"为假的那一半得不出结论。

对"Φ_x处处无定义"，在给定x之后，我们也可以先求出图灵机Z使$\#(Z)=x$，然后依次计算

$\Phi_{x,1}(0)$，$\Phi_{x,1}(1)$，

$\Phi_{x,2}(0)$, $\Phi_{x,2}(1)$, $\Phi_{x,2}(2)$,

$\Phi_{x,3}(0)$, $\Phi_{x,3}(1)$, $\Phi_{x,3}(2)$, $\Phi_{x,3}(3)$,

…………

来进行判断。如果事实上 Φ_x 处处无定义，上述过程不会完结，我们无法在有穷步内确认"Φ_x 处处无定义"为真这一事实；但如果"Φ_x 处处无定义"为假，则按上述过程作到某一步 t 时就会找到一个 z，$\Phi_{x,t}(z)\downarrow$，这时我们就可以断定"Φ_x 处处无定义"为假了。也就是说，对"Φ_x 处处无定义"，我们能确定它为假的那一半，但不能确定它为真的那一半，正好与 $\Phi_x(x)\downarrow$ 的情形相反。对于"$\Phi_x=\Phi_y$"，我们则无法确定其任何一半。因为要想肯定 $\Phi_x=\Phi_y$，我们需要确认

$$\Phi_x(0)=\Phi_y(0),\ \Phi_x(1)=\Phi_y(1),\ \Phi_x(2)=\Phi_y(2),\ \cdots$$

这是不可能在有穷步之内完成的。而要想确认 $\Phi_x\neq\Phi_y$，我们需要找到一个反例 k 使

$$\Phi_x(k)\neq\Phi_y(k),$$

如果 Φ_x 和 Φ_y 都是全函数，那么只要事实上 $\Phi_x\neq\Phi_y$，我们就总能找到这样的反例，但如果 Φ_x 和 Φ_y 中有一个定义域不全，就可能因 $\Phi_x(k)$ 或 $\Phi_y(k)$ 不能停机而无法确认 $\Phi_x\neq\Phi_y$。

为了区别上述三种情形，我们引入递归可枚举集和递归可枚举谓词的概念。

定义7.2.1 设 A 是自然数 n 元组的集合（$A\subseteq N^n$），如果 A 是某个 n 元递归函数的定义域，则称 A 为**递归可枚举集**。递归可枚举集简记作 ***r.e* 集**。

回忆第二章中我们曾用 $W_e^{(n)}$ 表示递归函数 $\Phi_e^{(n)}$ 的定义域，于是 n 维数组的递归可枚举集都包括在序列

$$W_0^{(n)},\ W_e^{(n)},\ \cdots,\ W_e^{(n)},\ \cdots \quad ①$$

中，而且每个 n 维的递归可枚举集都在序列①中出现无穷多次。

递归可枚举集的定义显示了**一种半可判定性**：对于任何一个 W_e，我们运行第 e 号图灵机图灵机，如果输入的 $x\in W_e$，则第 e 号

图灵机会停机；如果 $x \notin W_e$，则不会停机。因此，当 $x \in W_e$ 时，我们通过运行第 e 号图灵机就可以在有穷步之内确定 $x \in W_e$ 这一事实；但如果 $x \notin W_e$，靠运行第 e 号图灵机的办法就不能在有穷步之内得出任何结论。

定义7.2.2　设 R 是 n 元谓词，如果 R 的外延集

$$A=\{(x_1,\cdots,x_n)\mid R(x_1,\cdots,x_n)\text{为真}\}$$

是 $r.e$ 集，则称为**递归可枚举谓词**（也叫**半可判定谓词**）。递归可枚举谓词也简记为 ***r.e* 谓词**。

显然 A 是递归可枚举集 当且仅当 $(x_1,\cdots,x_n)\in A$ 是递归可枚举谓词。

例1　全体自然数的集合 N 是递归可枚举集，因为它是常函数

$$O(x)=0$$

的定义域。

例2　空集合 $\varnothing$ 是递归可枚举集，因为它是处处无定义的函数（空函数）的定义域。

例3　集合

$$K=\{x\mid \Phi_x(x)\text{有定义}\}$$

是递归可枚举集。因为它是一元递归函数 $\Phi_x(x)$ 的定义域。此外，由定理7.1.1我们已经知道 K 不是递归集。

定理7.2.1　$A\subseteq N^n$ 是递归可枚举集，当且仅当函数

$$f(x_1,\cdots,x_n)=\begin{cases}1 & \text{当}(x_1,\cdots,x_n)\in A,\\ \text{无定义} & \text{当}(x_1,\cdots,x_n)\notin A。\end{cases}$$

是递归函数。

证　设

$$f(x_1,\cdots,x_n)=\begin{cases}1 & \text{当}(x_1,\cdots,x_n)\in A,\\ \text{无定义} & \text{当}(x_1,\cdots,x_n)\notin A。\end{cases}$$

是递归函数，则 A 是 f 的定义域，根据定义7.2.1，A 是递归可枚举集。

反之，设 A 是 $r.e$ 集，则根据定义7.2.1，A 是某个 n 元递归函数 $g(x_1,\cdots,x_n)$的定义域。易见

$$f(x_1,\cdots,x_n)=Sg(g(x_1,\cdots,x_n)+1),$$

所以 f 是递归函数。

定理7.2.2　递归集都是递归可枚举集。

证　设 $A\subseteq N^n$ 是递归集，则谓词

$$(x_1,\cdots,x_n)\in A$$

是递归谓词，根据第三章习题12，函数

$$f(x_1,\cdots,x_n)=\begin{cases}1 & 当(x_1,\cdots,x_n)\in A,\\ 无定义 & 否则。\end{cases}$$

是递归函数，而 A 是 f 的定义域，所以 A 是递归可枚举集。

定理7.2.2的逆不成立。我们已经知道 $K=\{x|\Phi_x(x)\downarrow\}$就只是递归可枚举集而不是递归集(参看例3)。

递归集和递归可枚举集之间还有如下进一步的关系：

定理7.2.3　A 是递归集 当且仅当 A 和 $\overline{A}$ 都是递归可枚举集。

证　如果 A 是递归集，则根据定理3.2.4，$\overline{A}$ 也是递归集，由定理7.2.2，A 和 $\overline{A}$ 都是递归可枚举集。

如果 A 和 $\overline{A}$ 都是递归可枚举集，则根据定理7.2.1，函数

$$f(x_1,\cdots,x_n)=\begin{cases}1 & 当(x_1,\cdots,x_n)\in A,\\ 无定义 & 当(x_1,\cdots,x_n)\notin A。\end{cases}$$

和函数

$$g(x_1,\cdots,x_n)=\begin{cases}1 & 当(x_1,\cdots,x_n)\in \overline{A},\\ 无定义 & 当(x_1,\cdots,x_n)\notin \overline{A}。\end{cases}$$

都是递归函数。

设 Z_1和 Z_2分别是计算 f,g 的图灵机，对 Z_1和 Z_2输入$(x_1,\cdots,x_n)$，根据 f 和 g 的定义，Z_1和 Z_2中总有一部(也只有一部)会停机。如果是 Z_1停机，就表明$(x_1,\cdots,x_n)\in A$；如果是 Z_2停机，就表明$(x_1,\cdots,x_n)\in\overline{A}$。于是根据丘奇论题我们知道谓词

$$(x_1,\cdots,x_n)\in A$$

是递归谓词，从而集合 A 是递归集。

由于我们已经知道 K 是递归可枚举集但不是递归集，所以由定理立即得到

推论7.2.4　$\overline{K}=\{x\mid \Phi_x(x)\uparrow\}$不是递归可枚举集。

由此可知，递归可枚举集对补运算不封闭。因而根据定义7.2.2，递归可枚举谓词对否定也不封闭，例如谓词"$\Phi_x(x)$有定义"是递归可枚举的，而它的否定"$\Phi_x(x)$无定义"则不是递归可枚举的。

不过，递归可枚举集对并、交运算仍然是封闭的，即：

定理7.2.5　设 $A,B\subseteq N^n$，如果 A,B 都是递归可枚举集，则 $A\cup B$和 $A\cap B$ 也都是递归可枚举集。

证　设 $A,B\subseteq N^n$ 且 A,B 都是递归可枚举集。由定理7.2.1，

$$f(x_1,\cdots,x_n)=\begin{cases}1 & \text{当}(x_1,\cdots,x_n)\in A,\\ \text{无定义} & \text{当}(x_1,\cdots,x_n)\notin A。\end{cases}$$

和

$$g(x_1,\cdots,x_n)=\begin{cases}1 & \text{当}(x_1,\cdots,x_n)\in B,\\ \text{无定义} & \text{当}(x_1,\cdots,x_n)\notin B。\end{cases}$$

都是递归函数。

设 $\Phi_e^{(n)}=f$，$\Phi_k^{(n)}=g$，定义函数

$$h(x_1,\cdots,x_n)=\begin{cases}1 & \text{当}(x_1,\cdots,x_n)\in A\text{ 或}\\ & \quad(x_1,\cdots,x_n)\in B,\\ \text{无定义} & \text{否则。}\end{cases}$$

我们可以按如下直观算法计算 $h(x_1,\cdots,x_n)$：

同时计算 $\Phi_e^{(n)}(x_1,\cdots,x_n)$和 $\Phi_k^{(n)}(x_1,\cdots,x_n)$，如果其中有一部停机，则 $h(x_1,\cdots,x_n)=1$，否则运算将永不停止，即 $h(x_1,\cdots,x_n)$无定义。根据丘奇论题，h 是递归函数。容易看出 $A\cup B$ 是 h 的定义域，所以 $A\cup B$ 是递归可枚举集。

$A\cap B$ 的情形是类似的，留给读者

此外，递归可枚举集对投影运算也是封闭的，即：

定理7.2.6　如果$n+1$元数组集A是递归可枚举集，则A的投影集

$$B=\{(x_1,\cdots,x_n)|\exists y((x_1,\cdots,x_n,y)\in A)\}$$

也是递归可枚举集。

证　设$n+1$元数组集A是递归可枚举集，则由定理7.2.1，函数

$$f(x_1,\cdots,x_n,y)=\begin{cases}1 & 当(x_1,\cdots,x_n,y)\in A,\\ 无定义 & 当(x_1,\cdots,x_n,y)\notin A。\end{cases}$$

是递归函数。令e是计算f的图灵机($\Phi_e=f$)，我们可以用如下算法计算函数

$$g(x_1,\cdots,x_n)=\begin{cases}1 & 当(x_1,\cdots,x_n)\in B,\\ 无定义 & 当(x_1,\cdots,x_n)\notin B;\end{cases}$$

依次计算：

$\Phi_{e,1}(x_1,\cdots,x_n,0)$；

$\Phi_{e,2}(x_1,\cdots,x_n,0)$，　$\Phi_{e,2}(x_1,\cdots,x_n,1)$；

$\Phi_{e,3}(x_1,\cdots,x_n,0)$，　$\Phi_{e,3}(x_1,\cdots,x_n,1)$，　$\Phi_{e,3}(x_1,\cdots,x_n,2)$；

…………

如果计算到某一步，其中有一项(比如$\Phi_{e,t}(x_1,\cdots,x_n,k)$停机，则$g(x_1,\cdots,x_n)$等于1，否则$g(x_1,\cdots,x_n)$无定义。

根据丘奇论题，g是递归函数，从而由定理7.2.1，B是递归可枚举集。

由上面两个定理和定义7.2.2，我们容易得到

推论7.2.7　(1) 递归可枚举谓词对$\wedge$、$\vee$封闭，即如果P,Q都是n元递归可枚举集谓词，则$P\wedge Q$和$P\vee Q$也都是递归可枚举谓词。

(2) 递归可枚举谓词对存在量词封闭，即如果P是$n+m$元递归可枚举谓词，则

$\exists y_1\cdots\exists y_m P(x_1,\cdots,x_n,y_1\cdots y_m)$也是($n$元)递归可枚举谓词。

由于对任何的正整数 n，n 元配对函数 J_n 是原始递归函数，所以 n 元组集合 A 是递归可枚举集(递归集)当且仅当 A 在 J_n 下的象集

$$J_n[A]=\{J_n(x_1,\cdots,x_n)\mid(x_1,\cdots,x_n)\in A\}$$

是递归可枚举集(递归集)。

因此在讨论递归可枚举集(递归集)的时候我们可以把 n 元数组 $(x_1,\cdots,x_n)$ 与它在配对函数 J_n 下的值 $J_n(x_1,\cdots,x_n)$ 等同看待；把 n 元组集合 A 和 A 在 J_n 下的象集 $J_n[A]$ 等同看待。从而，如果没有特殊需要，今后我们在讨论递归可枚举集(递归集)时就只限于讨论自然数集。

定理7.2.8 如果 $f(x_1,\cdots,x_n)$，$g(x_1,\cdots,x_n)$ 都是递归函数，则 n 元谓词

$$f(x_1,\cdots,x_n)=g(x_1,\cdots,x_n)$$

是 n 元递归可枚举谓词。

证 令

$$h(x_1,\cdots,x_n)=\begin{cases}1 & \text{当 } f(x_1,\cdots,x_n)=g(x_1,\cdots,x_n)，\\ \uparrow & \text{否则。}\end{cases}$$

容易验证

$$h(x_1,\cdots,x_n)=\overline{Sg}(C_=(f(x_1,\cdots,x_n),g(x_1,\cdots,x_n))-1)。 \quad ①$$

因为当 $f(x_1,\cdots,x_n)$ 或 $g(x_1,\cdots,x_n)$ 无定义时，

$$\overline{Sg}(C_=(f(x_1,\cdots,x_n),g(x_1,\cdots,x_n))-1)$$

也无定义；当 $f(x_1,\cdots,x_n)$ 和 $g(x_1,\cdots,x_n)$ 都有定义但

$$f(x_1,\cdots,x_n)\neq g(x_1,\cdots,x_n)$$

时，

$$\overline{Sg}(C_=(f(x_1,\cdots,x_n),g(x_1,\cdots,x_n))-1)$$

仍无定义；当 $f(x_1,\cdots,x_n)$ 和 $g(x_1,\cdots,x_n)$ 都有定义且

$$f(x_1,\cdots,x_n)=g(x_1,\cdots,x_n)$$

时，

$$\overline{Sg}(C_=(f(x_1,\cdots,x_n),g(x_1,\cdots,x_n))-1)=1。$$

所以①成立，$h(x_1,\cdots,x_n)$是递归函数。根据定理7.2.1和定义7.2.2，

$$f(x_1,\cdots,x_n)=g(x_1,\cdots,x_n)$$

是递归可枚举谓词。

定理7.2.9　令 $A\subseteq N$，以下四个命题彼此等价：

(1) A 是 $r.e$ 集，

(2) $A=\varnothing$或 A 是某个一元原始递归函数 f 的值域，

(3) $A=\varnothing$或 A 是某个一元递归全函数 f 的值域，

(4) A 是某个一元递归函数的值域。

证　(2)⇒(3)和(3)⇒(4)是显然的，以下证明(4)⇒(1)和(1)⇒(2)。

先证(4)⇒(1)。设 $A=\mathrm{ran}\ g$，g 是递归函数。则由定理7.2.9知

$$g(x)=y$$

是递归可枚举谓词，而

$$y\in A\Leftrightarrow\exists x(g(x)=y)$$

于是根据推论7.2.7之(2)"$y\in A$"是递归可枚举谓词，也就是说 A 是递归可枚举集。

再证(1)⇒(2)。设 A 是递归可枚举集且 $A\neq\varnothing$。则由定理7.2.1，

$$h(x)=\begin{cases}1 & \text{当 } x\in A,\\ \text{无定义} & \text{当 } x\notin A\end{cases}$$

是递归函数。于是对某个 e，$h=\Phi_e$，根据范式定理(定理4.2.1)我们有

$$x\in A\Leftrightarrow\exists yT(x,y,e)\qquad ①$$

由于 $A\neq\varnothing$，设 a 是 A 中的最小元，定义函数 f

$$\begin{cases}f(0)=a,\\ f(z+1)=C_{\neg T}(l(z+1),r(z+1))\cdot f(z)\\ \qquad\qquad +C_T(l(z+1),r(z+1))\cdot l(z+1)\end{cases}$$

其中 C_T 和 $C_{\neg T}$ 分别是 $T(x,y,e)$ 和 $\neg T(x,y,e)$ 的特征函数，它们是关于 x,y 的二元函数，e 是常数。

f 显然是原始递归函数。我们来证明 $A=\mathrm{ran}\ f$。

设 $x\in A$，由①知存在(唯一的)y 使 $T(x,y,e)$ 为真(即 $C_T(x,y)=1$，$C_{\neg T}(x,y)=0$)。根据编号的方法可知 $y\neq 0$，所以 $J(x,y)\neq 0$。令 $J(x,y)=z+1$(从而 $x=l(z+1)$，$y=r(z+1)$)，由 f 的定义可知

$$f(z+1)=l(z+1)=x,$$

于是 $x\in\mathrm{ran}\ f$，因而 $A\subseteq\mathrm{ran}\ f$。

另一方面，我们用归纳法来证明 $\mathrm{ran}\ f\subseteq A$。

首先，$f(0)=a\in A$。

其次，设 $f(z)\in A$。由于 C_T 和 $C_{\neg T}$ 的值总有一个为0，于是，当 $C_T(l(z+1),r(z+1))=0$ 时，

$$f(z+1)=f(z)\in A;$$

当 $C_{\neg T}(l(z+1),r(z+1))=0$ 时，

$$f(z+1)=l(z+1),$$

而由于此时 $C_T(l(z+1),r(z+1))=1$，即 $T(l(z+1),r(z+1),e)$ 为真，由①可得 $l(z+1)\in A$ 从而 $f(z+1)\in A$，这就得到

$$\mathrm{ran}\ f\subseteq A。$$

定理7.2.9之(2)，(3)中所说的递归全函数(原始递归函数)f 称为(非空的)$r.e$ 集 A 的**枚举函数**。它表明，非空的递归可枚举集可以用一个递归全函数逐个枚举出来，即

$$A=\{f(0),f(1),f(2),\cdots f(n),\cdots\}。$$

我们用 E_e 表示 Φ_e 的值域，于是每个递归可枚举集就都在序列

$$E_0,\ E_1,\ \cdots,\ E_e,\ \cdots$$

中出现无穷多次。

非空递归可枚举集的枚举函数不是唯一的，事实上每个非空的递归可枚举集 A 都有无穷多个枚举函数，而且在枚举过程中可

能有元素重复出现(如果 A 是有穷集则必定重复)。

定理7.2.11 如果 A 是无穷的递归可枚举集,则 A 有1-1的枚举函数。

证 设 g 是 A 的一个枚举函数

$$A=\{g(0),\ g(1),\ \cdots,g(n),\ \cdots\}$$

自左向右删去重复的项,就得到一个1－1的枚举。形式地说,定义函数 f 为:

$$\begin{cases} f(0)=g(0), \\ f(y+1)=g(\mu z[\overset{y}{\forall} u(f(u)\neq g(z))]), \end{cases}$$

f 就是所要的1-1的枚举函数(由于 A 是无穷集,故 f 是全函数)。

定理7.2.12 设 A 是无穷的递归可枚举集,A 是递归集当且仅当 A 有一个严格上升的枚举函数。

证 设 A 是一个无穷的递归集,则函数

$$\begin{cases} f(0)=\mu u[u\in A] \\ f(y+1)=\mu u[u\in A\wedge u>f(y)] \end{cases}$$

是 A 的一个严格上升的枚举函数。

反之,设 A 有一个严格上升的枚举函数 f。由 f 严格上升知:

如果 $f(x)=y$,则 $x\leqslant y$。

于是

$$\begin{aligned} y\in A &\Leftrightarrow \exists x(f(x)=y) \\ &\Leftrightarrow \overset{y}{\exists} x(f(x)=y) \end{aligned}$$

由 f 是全函数,故 $f(x)=y$ 是递归谓词,从而 A 是递归集。

定理7.2.13 无穷的递归可枚举集都有无穷的递归子集。

证 设 A 是一个无穷的递归可枚举集,g 是它的一个枚举函数,

$$A=\{g(0),\ g(1),\cdots\}$$

由于 A 是无穷的,所以我们可以从 $g(0)$, $g(1)$,…中能行地选出一个严格上升的子序列,形式地说就是:定义函数 f 如下:

$$\begin{cases} g(0)=f(0), \\ g(y+1)=g(\mu u[f(y)<g(u)]); \end{cases}$$

由于 A 无穷，故 f 是全函数，利用归纳法容易验证 f 是严格上升的；由 f 的定义容易看出

$$\text{ran } f \subseteq A。$$

根据定理7.2.12，ran f 是 A 的一个递归子集。

递归函数与递归可枚举集(递归可枚举谓词)之间还有如下的一些关系：

定理7.2.14

(1) 如果 f 是 n 元递归函数，$A \subseteq N^n$ 是递归可枚举集，则 A 在 f 下的像集

$$f[A] =_{df} \{f(x_1,\cdots,x_n) \mid (x_1,\cdots,x_n) \in A\}$$

是递归可枚举集。

(2) 如果 $P(x_1,\cdots,x_n,y)$是递归可枚举谓词，$f(x_1,\cdots,x_n)$是递归全函数，则

$$P(x_1,\cdots,x_n,f(x_1,\cdots,x_n))$$

是递归可枚举谓词。

(3)(图象定理) n 元函数 f 是递归函数 当且仅当 f 的图象

$$G_f =_{df} \{(x_1,\cdots,x_n,y) \mid f(x_1,\cdots,x_n)=y\}$$

是递归可枚举集。

证 对于(1)，由于

$$f[A]=\{y \mid \exists x_1\cdots\exists x_n((x_1,\cdots,x_n)\in A \wedge y=f(x_1,\cdots,x_n))\}$$

因为 A 是 $r.e$ 集，故$(x_1,\cdots,x_n)\in A$ 是 $r.e$ 谓词；由定理7.2.8知 $y=f(x_1,\cdots,x_n)$ 是 $r.e$ 谓词，从而由7.2.7之(2)知

$$\exists x_1\cdots\exists x_n((x_1,\cdots,x_n)\in A \wedge y=f(x_1,\cdots,x_n))$$

是递归可枚举谓词，于是 $f[A]$是递归可枚举集。

对于(2)，由于

$$P(x_1,\cdots,x_n,f(x_1,\cdots,x_n)) \Leftrightarrow \exists y(P(x_1,\cdots,x_n,y) \wedge \wedge y=f(x_1,\cdots,x_n)),$$

由定理7.2.8和推论7.2.7知右端的公式是递归可枚举谓词。

对于(3),由于

$$(x_1,\cdots,x_n,y)\in G_f \Leftrightarrow y=f(x_1,\cdots,x_n),$$

由定理7.2.8即得到 G_f 是递归可枚举集。

习　　题

1. 证明下述谓词都是非递归的:

(1) $x\in E_x$,

(2) Φ_x 为全函数,

(3) $W_x=W_y$,

(4) $\Phi_x(x)=0$,

(5) $\Phi_x(y)=0$,

(6) $x\in E_y$,

(7) Φ_x 是常函数,

(8) Φ_x 处处无定义(即 $W_x=\varnothing$),

(9) W_x 是无穷集,

(10) E_x 是无穷集,

(11) $c\in W_x$(c 是常数)。

2. 证明:集合$\{x\mid\Phi_x$ 不是1-1的$\}$是 $r.e$ 集。

3. 证明以下谓词都是递归可枚举的:

(1) $W_x\neq\varnothing$;

(2) $E_x\neq\varnothing$;

(3) $\Phi_x(x)$是完全平方数;

4. 证明:如果 A 是 $r.e$ 集,则 $\bigcup_{x\in A}W_x$ 和 $\bigcup_{x\in A}E_x$ 都是 $r.e$ 集,但 $\bigcap_{x\in A}W_x$ 未必是 $r.e$ 集。

5. 证明:存在递归全函数 $k(x)$和 $m(x)$,使对每个 x 有 $W_x=E_{k(x)}$以及 $E_x=W_{m(x)}$。

第八章

m-归约和m-度

上一章我们引入了递归可枚举集和递归可枚举谓词并实际证明了一些集合和谓词是递归可枚举的。证明所使用的方法可以分为两类：由定义直接证明（例如定理7.2.1的证明）和将一个集合（谓词）的递归可枚举性归约为另一个集合（谓词）的递归可枚举性。在这一章，我们对归约方法作一些讨论。

§8.1 m-归约

定义8.1.1 设 $A,B\subseteq N$，如果存在一个递归全函数 f 使

$$x\in A \Leftrightarrow f(x)\in B$$

（或者等价地，$f[A]\subseteq B$ 且 $f[\overline{A}]\subseteq\overline{B}$），则称 A 可以**多一归约**（**m-归约**）到 B，记作

$$A\leqslant_m B,$$

函数 f 称为（从 A 到 B 的）**归约函数**。如果 f 是1-1的，则称 A 可以**一一归约**到 B 记作

$$A\leqslant_1 B。$$

由定义容易看出：

如果 $A\leqslant_1 B$，则 $A\leqslant_m B$。

但反向的蕴涵不成立。例如对集合 $A=\{1,2,3\}$，$B=\{0,1\}$，递归全函数

$$f(x)=\begin{cases}0 & 当\ x\in A,\\7 & 当\ x\notin A\end{cases}$$

将 A m-归约到 B,但 A 不可能一一归约到 B。以下我们主要讨论 m-归约。

在第七章中,我们实际上已经证明了一些 m-归约关系,例如

$K\leqslant_m\{x\mid\Phi_x=O\}$, (定理7.1.3)

$K\leqslant_m\{x\mid c\in E_x\}$, (定理7.1.5)

$K\leqslant_m\{x\mid c\in W_x\}$, (定理7.1.6)

等等。以下我们来看一下 m-归约的基本性质:

定理8.1.1 $\leqslant_m$ 具有以下基本性质:

(1) 对任何集合 $A\leqslant_m A$(自反性);

(2) 如果 $A\leqslant_m B$ 且 $B\leqslant_m C$,则 $A\leqslant_m C$(传递性);

(3) $A\leqslant_m B$ 当且仅当 $\overline{A}\leqslant_m\overline{B}$;

(4) 如果 B 是递归集且 $A\leqslant_m B$,则 A 也是递归集;

(5) 如果 B 是 $r.e$ 集且 $A\leqslant_m B$,则 A 也是 $r.e$ 集;

(6) 如果 A 是递归集,$B\neq\varnothing,N$,则 $A\leqslant_m B$;

(7) $A\leqslant_m N$ 当且仅当 $A=N$,

$A\leqslant_m\varnothing$ 当且仅当 $A=\varnothing$;

(8) $N\leqslant_m A$ 当且仅当 $A\neq\varnothing$,

$\varnothing\leqslant_m A$ 当且仅当 $A\neq N$;

(上述性质除(6)和(8)外,对$\leqslant_1$也都成立。)

证 只证(1),(5),(6),其余留给读者。

(1) 恒同函数 $f(x)=x$ 是将 A m-归约到 A 的归约函数。

(5) 设 $B=W_e$,h 是从 A 到 B 的归约函数,则 A 是复合函数 $\Phi_e(h(x))$的定义域。

(6) 由于 A 是递归集,所以 $\overline{A}$ 也是递归集,$x\in A$ 和 $x\notin A$ 都是递归谓词。由于 $B\neq\varnothing,N$,所以 $\overline{B}\neq N,\varnothing$。取 $b\in B$,$c\in\overline{B}$。则函数

$$f(x)=\begin{cases}b & \text{当 } x\in A,\\ c & \text{当 } x\notin A\end{cases}$$

是递归全函数。易见

$$x \in A \Leftrightarrow f(x)=b \in B。$$

推论8.1.2　如果 A 是非递归的递归可枚举集，则 $\overline{A} \not\leqslant_m A$ 且 $A \not\leqslant_m \overline{A}$。

证　若 $\overline{A} \leqslant_m A$，则因 A 是递归可枚举集，由定理之(5)即得 $\overline{A}$ 是递归可枚举集，从而 A 是递归集，矛盾，所以 $\overline{A} \not\leqslant_m A$。

由定理8.1.1之(3)

$$A \leqslant_m \overline{A} \Leftrightarrow \overline{A} \leqslant_m A,$$

于是由 $\overline{A} \not\leqslant_m A$ 得到 $A \not\leqslant_m \overline{A}$。

满足 $A \leqslant_m \overline{A}$ 的集合 A 称为**自对偶的**，推论8.1.2 表明不具备自对偶性是非递归的递归可枚举集的一个特征。

定义8.1.2　如果 A 是递归可枚举集，而且对任何递归可枚举集 B 都有 $B \leqslant_m A$，则称 A 是 **m-完全集**。

由定义立即得出：若 A, B 都是 m-完全的，则 $A \leqslant_m B$ 且 $B \leqslant_m A$。

定理8.1.3　$K=\{x \mid \Phi_x(x)\downarrow\}$是 m-完全的，即对任何集合 A，A 是 $r.e$ 集当且仅当 $A \leqslant_m K$。

证　由于我们已经知道 K 是递归可枚举集，所以只须证明对任何递归可枚举的 A 都有 $A \leqslant_m K$。考虑函数

$$g(x,y)=\begin{cases}1 & \text{当 } x \in A \\ \text{无定义} & \text{当 } x \notin A\end{cases}$$

由 A 是 $r.e$ 集知 g 是(二元)递归函数，根据 s-m-n 定理，存在递归全函数 $t(x)$使

$$g(x,y)=\Phi_{t(x)}(y)。$$

由 g 的定义知当 $x \in A$ 时，$\Phi_{t(x)}$作为 y 的一元函数是常函数，因而是全函数；当 $x \notin A$ 时，$\Phi_{t(x)}$作为 y 的一元函数是处处无定义的函数。于是

$$x \in A \Leftrightarrow \Phi_{t(x)}\text{处处有定义} \Leftrightarrow \Phi_{t(x)}(t(x))\downarrow \Leftrightarrow t(x) \in K。$$

所以 $A \leqslant_m K$。

定义8.1.2为我们提供了一个证明一个集合是递归可枚举集

的方法：要证明一个集合 A 是 $r.e$ 集，只要证明$A\leqslant_m K$；而要证明 $A\leqslant_m K$，就要找到一个递归全函数 $t(x)$，使

$$x\in A \Leftrightarrow t(x)\in K。$$

找出 $t(x)$则经常是依据 s-m-n 定理。

同样，要证明一个集合 B 不是递归可枚举的，我们可以使用证明 $\overline{K}\leqslant_m B$ 的方法：只要找到一个递归全函数 $t(x)$使得

$$x\notin K \Leftrightarrow t(x)\in B,$$

就表明 B 不是递归可枚举集。找到这个 $t(x)$的方法也常是使用 s-m-n 定理。

此外，由上述定义和定理8.1.3我们也可以建立一种证明一个递归可枚举集是 m-完全集的方法，那就是

推论8.1.4　如果 A 是 $r.e$ 集，则

A 是 m-完全的 当且仅当 $K\leqslant_m A$。

定义8.1.3　如果 $A\leqslant_m B$ 且 $B\leqslant_m A$，则称 A 和 B 是 m-等价的，记作

$$A\equiv_m B。$$

例1　设 $K_0=\{(x,y)\mid\Phi_x(y)\downarrow\}=\{(x,y)\mid y\in W_x\}$。则

$K\equiv_m K_0$。

证　我们需要证明(1) $K_0\leqslant_m K$，及(2) $K\leqslant_m K_0$。

对于(1)，根据第五章习题2，函数

$$f(x,y,z)=\begin{cases}0 & \text{如果 } \Phi_x(y)\downarrow\\ \uparrow & \text{否则}\end{cases}$$

是递归函数。于是由定理7.2.1知 K_0是 $r.e$ 集。由定理8.1.3知 $K_0\leqslant_m K$。

对于(2)，由于

$$x\in K \Leftrightarrow (x,x)\in K_0,$$

自然有 $K\leqslant_m K_0$。

例2　设 $K_1=\{x\mid W_x\neq\varnothing\}$，则

$K_1\equiv_m K$。

证 根据第五章习题3及定理7.2.1和定理8.1.2知 $K_1 \leqslant_m K$。为了证明 $K \leqslant_m K_1$，利用 s-m-n 定理定义函数 $t(x)$，使得

$$\Phi_{t(x)}(y) = \begin{cases} 1 & \text{如果 } x \in W_x, \\ \uparrow & \text{否则。} \end{cases}$$

于是

$x \in W_x \Rightarrow \Phi_{t(x)}$是常函数 $\Rightarrow W_{t(x)} \neq \varnothing$；

$x \notin W_x \Rightarrow \Phi_{t(x)}$处处无定义 $\Rightarrow W_{t(x)} = \varnothing$。

因此，$K_1 \equiv_m K$。

例1和例2表明 K_0和 K_1也都是 m-完全的。

此外，由定义和定理8.1.1还容易得出以下基本性质：

(1) 一切 m-完全集彼此 m-等价；

(2) 一切既不等于$\varnothing$也不等于 N 的递归集彼此 m-等价；

(3) $\varnothing$，N 只与其自身 m-等价；

(4) $\equiv_m$ 是等价关系(满足自反、对称、传递三个性质)。

定义8.1.4 自然数集在$\equiv_m$关系下的等价类称为 m-**度**，含有集合 A 的 m-度记作 $\deg_m(A)$，也记作 $\bar{a}$，即

$$\bar{a} = \deg_m(A) = \{B \mid B \equiv_m A\}。$$

设 $\bar{a}$ 和 $\bar{b}$ 是两个 m-度，如果 $A \in \bar{a}$，$B \in \bar{b}$ 且 $A \leqslant_m B$，则称 m-度 $\bar{a}$ 小于等于 $\bar{b}$，记作

$$\bar{a} \leqslant \bar{b}。$$

如果 $\bar{a} \leqslant \bar{b}$ 且 $\bar{a} \neq \bar{b}$，则称 $\bar{a}$小于 $\bar{b}$，记作 $\bar{a} < \bar{b}$。

显然 m-度上的$\leqslant$关系是个偏序关系(自反、传递)；m- 度上的$<$关系是个严格偏序关系(反自反、传递)。

根据前面的结果，容易得到以下结论：

1. $\deg_m(\varnothing) = \{\varnothing\}$ （这个 m-度记作 **o**)；
2. $\deg_m(N) = \{N\}$ （这个 m-度记作 **n**)；
3. 如果 A 是递归集且 $A \neq \varnothing, N$，则 $\deg_m(A)$恰由全体既非$\varnothing$又非 N 的递归集组成 （这个 m-度记作 0_m)；
4. $\deg_m(K)$恰由全体 m-完全集组成(这个 m-度记作 $0_m'$)；

5. $O_m < O_m'$；

6. 任给 m-度 $\bar{a}$，如果 $\bar{a} \neq$ **o**，**n**，则 **o** $< \bar{a}$，**n** $< \bar{a}$（即 **o**，**n** 是两个极小的 m-度）。

8. **o** 和 **n** 不可比较，即 **o** $\not\leqslant$ **n** 且 **n** $\not\leqslant$ **o**。

m-度的结构可以用如下的图形来表示：

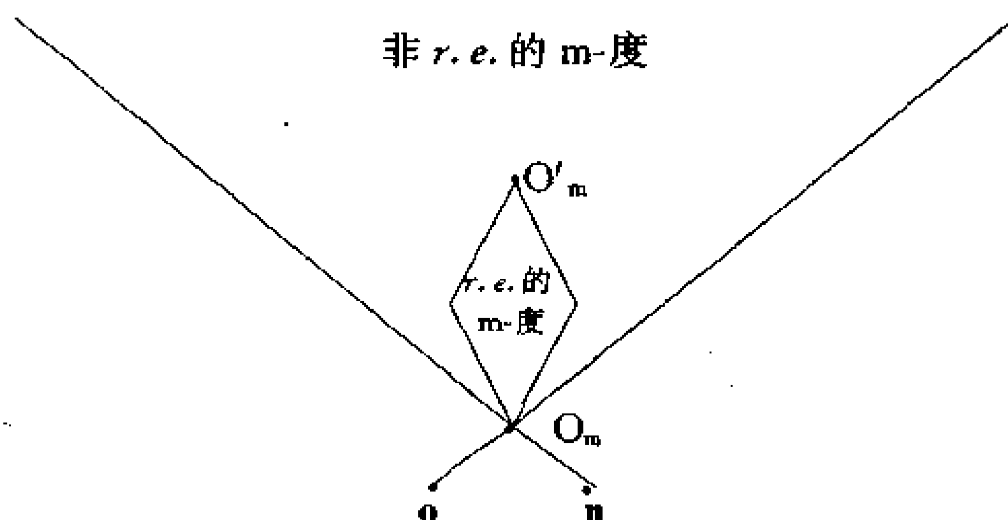


图8.1 m-度

§8.2 指 标 集

前面我们已经看到，在算术中所谈论的各种自然数集都是递归集，我们至今所接触到的非递归集都是借助递归论中的方法建立的。现在，我们对上面所使用过的定义非递归集的方法做一点系统的讨论。

定义8.2.1 设 $\mathscr{R}$ 是全体（一元）递归函数的类，$\mathscr{A} \subseteq \mathscr{R}$，集合

$$A = \{x \mid \Phi_x \in \mathscr{A}\}$$

称为**指标集**。

指标集的基本特征是：如果 $x\in A$ 且 $\Phi_y=\Phi_x$，则 $y\in A$。例如下述集合都是指标集：

(1) $K_1=\{x|W_x\neq\varnothing\}$，

(2) $\overline{K_1}=\{x|W_x=\varnothing\}$，

(3) $\mathrm{Fin}=\{x|W_x$ 有穷$\}$，

(4) $\mathrm{Inf}=\overline{\mathrm{Fin}}=\{x|W_x$ 无穷$\}$，

(5) $\mathrm{Tot}=\{x|\Phi_x$ 是全函数$\}=\{x|W_x=N\}$，

(6) $\overline{\mathrm{Tot}}=\{x|\Phi_x$ 不是全函数$\}=\{x|W_x\neq N\}$，

(7) $\mathrm{Con}=\{x|\Phi_x$ 是常函数$\}$，

(8) $\mathrm{Cof}=\{x|\overline{W_x}$有穷$\}$，

(9) $\mathrm{Rec}=\{x|W_x$ 是递归集$\}$。

由定义容易看出 A 是指标集当且仅当 $\overline{A}$ 是指标集。

下面我们来看一下指标集的性质和指标集的 m-度之间的关系。

定理8.2.1(**赖斯定理**) 设 A 是指标集，如果 $A\neq\varnothing,N$，则 A 不是递归集。

证 设 $A=\{x|\Phi_x\in\mathscr{A}\}$。我们不妨设处处无定义的函数 $f_\varnothing\notin\mathscr{A}$，否则我们可以用 $\overline{A}$ 代替 A 进行讨论，因为 A 是递归集当且仅当 $\overline{A}$ 是递归集。

由于 $A\neq\varnothing$ 且 $f_\varnothing\notin\mathscr{A}$，故有函数 g 使 $g\in\mathscr{A}$ 且 $g\neq f_\varnothing$。任取一个这样的 g，定义函数

$$f(x,y)=\begin{cases}g(y) & \text{如果 } x\in W_x,\\ \text{无定义} & \text{如果 } x\notin W_x。\end{cases}$$

显然 f 是二元递归函数。于是由 s-m-n 定理存在递归全函数 $t(x)$ 使

$$f(x,y)=\Phi_{t(x)}(y),$$

从而

$$x\in W_x\Rightarrow\Phi_{t(x)}=g\Rightarrow\Phi_{t(x)}\in\mathscr{A}\ ;$$

$$x \notin W_x \Rightarrow \Phi_{t(x)} = f_{\varnothing} \Rightarrow \Phi_{t(x)} \notin \mathscr{A}\ ;$$

即 $$x \in W_x \Leftrightarrow t(x) \in A。$$

也就是说 $K \leqslant_m A$。因此，对于非 N 非$\varnothing$的指标集 A，或者 $K \leqslant_m A$ 或者 $K \leqslant_m \overline{A}$，总之 A 和 $\overline{A}$ 都不是递归集。

定理8.2.1为我们揭示了指标集的这样的性质：

根据定理8.1.1之(4)可知如果指标集 $A \neq \varnothing, N$，则 A 不是递归集。换句话说，指标集中只有$\varnothing$，N 两个递归集。

那么究竟哪些指标集是递归可枚举的呢？赖斯的另一个定理给出了一个必要条件。

我们称定义域有穷的函数为**有穷函数**。对两个函数 f 和 g，如果 $\mathrm{dom}\ f \subseteq \mathrm{dom}\ g$ 并且当 $x \in \mathrm{dom} f$ 时有 $f(x) = g(x)$，则称函数 f 是函数 g 的**子函数**，记作 $f \subseteq g$。

定理8.2.1(**赖斯-沙皮罗定理**)　设 $A = \{x \mid \Phi_x \in \mathscr{A}\}$是指标集，如果 A 是 $r.e$ 集，则对任何递归函数 f：

$$f \in \mathscr{A} \text{　当且仅当存在有穷函数 } \theta \subseteq f \text{ 使 } \theta \in \mathscr{A}\ 。$$

证　设 $A = \{x \mid \Phi_x \in \mathscr{A}\}$是 $r.e$ 集，$f \in \mathscr{A}$，但对任何有穷的 $\theta \subseteq f$，$\theta \notin \mathscr{A}$。

定义(二元)函数

$$g(x,t) = \begin{cases} f(t) & \text{如果 } \Phi_{x,t}(x) = 0, \\ \text{无定义} & \text{如果 } \Phi_{x,t}(x) = 1。\end{cases}$$

根据 $\Phi_{x,t}(x)$的定义(见§5.1例1)，$\Phi_{x,t}(x)$是递归全函数，于是 g 是递归函数。由 s-m-n 定理，存在递归全函数 $s(x)$使得

$$g(x,t) = \Phi_{s(x)}(t)。$$

由 g 的定义容易看出，对每个固定的 x，$g(x,t)$作为 t 的一元函数是 f 的子函数。即对每个固定的 x，$\Phi_{s(x)} \subseteq f$。于是

$$x \in W_x \Rightarrow \Phi_x(x)\downarrow \Rightarrow \text{存在 } t_0\text{，使当 } t \geqslant t_0 \text{ 时 } \Phi_{x,t}(x) = 1$$
$$\Rightarrow \Phi_{s(x)}\text{有穷} \Rightarrow \Phi_{s(x)} \notin \mathscr{A} \Rightarrow s(x) \notin A;$$
$$x \notin W_x \Rightarrow \Phi_x(x)\uparrow \Rightarrow \text{对任何 } t\text{，} \Phi_{x,t}(x) = 0 \Rightarrow \Phi_{s(x)} = f$$
$$\Rightarrow \Phi_{s(x)} \in \mathscr{A} \Rightarrow s(x) \in A。$$

也就是说 $\overline{K}\leqslant_m A$。而我们已经知道 $\overline{K}$ 不是 $r.e$ 集，矛盾。这个矛盾就表明我们的假设是不成立的，因此如果 $A=\{x|\Phi_x\in\mathscr{A}\}$ 是 $r.e$ 集，$f\in\mathscr{A}$，则存在有穷的 $\theta\subseteq f$，使 $\theta\in\mathscr{A}$。

反之，设 $A=\{x|\Phi_x\in\mathscr{A}\}$ 是 $r.e$ 集，存在有穷的 $\theta\subseteq f$ 使 $\theta\in\mathscr{A}$，但 $f\notin\mathscr{A}$。

定义函数

$$h(x,t)=\begin{cases} f(t) & \text{如果 } t\in\mathrm{dom}\ \theta \text{ 或 } \Phi_x(x)\downarrow, \\ \text{无定义} & \text{否则。}\end{cases}$$

容易验证 $h(x,t)$ 是（二元）递归函数。由 s-m-n 定理，存在递归全函数 $s(x)$ 使

$$h(x,t)=\Phi_{s(x)}(t)。$$

注意到 h 的定义保证 $\theta\subseteq\Phi_{s(x)}$。于是有

$$x\in W_x\Rightarrow\Phi_x(x)\downarrow\ \Rightarrow\Phi_{s(x)}=f\Rightarrow\Phi_{s(x)}\notin\mathscr{A}\ \Rightarrow s(x)\notin A;$$

$$x\notin W_x\Rightarrow\Phi_x(x)\uparrow\ \Rightarrow\Phi_{s(x)}=\theta\Rightarrow\Phi_{s(x)}\in\mathscr{A}\ \Rightarrow s(x)\in A。$$

也就是说 $\overline{K}\leqslant_m A$。仍与 A 是 $r.e$ 集矛盾。因此如果 $A=\{x|\Phi_x\in\mathscr{A}\}$ 是 $r.e$ 集且存在有穷的 $\theta\subseteq f$ 使 $\theta\in\mathscr{A}$，则 $f\in\mathscr{A}$。

利用定理8.2.1可以证明一些集合不是递归可枚举集。例如对指标集

$$\mathrm{Tot}=\{x|W_x=N\},$$

由于它由全函数的指标组成，因此不可能含有有穷函数的指标，于是由定理8.2.1可知 Tot 不是 $r.e$ 集。

再如指标集

$$\overline{\mathrm{Tot}}=\{x|W_x\neq N\},$$

它由全体定义域不全的递归函数的指标组成，其中自然包括全体有穷函数的指标。对任何一个递归全函数 f，如果 $\theta\subseteq f$ 且 θ 有穷，则 θ 的指标都在 $\overline{\mathrm{Tot}}$ 中，但 f 的指标却不在 A 中。因此由定理8.2.1，$\overline{\mathrm{Tot}}$ 不是 $r.e$ 集。

在这一节的最后，我们来总结一下寻找归约函数的方法。从上面两节的证明中我们看到寻找使 $A\leqslant_m B$ 的归约函数的过程可以

分成三个步骤：

1. 先定义一个与 A 有关的递归函数 $g(x,y)$；

2. 利用 s-m-n 定理得到一个递归全函数 $t(x)$，使

$$g(x,y)=\Phi_{t(x)}(y);$$

3. 证明 $x\in A$ 当且仅当 $t(x)\in B$。

三个步骤中的关键是第一步，后面的两步都是由第一步保证的：要使用 s-m-n 定理，就要求所定义的函数必须是递归的；而要得到第三条，也必须要求所定义的函数能保证在 $x\in A$ 时 $g(x,y)$ 作为 y 的一元函数要满足 B 中的条件；在 $x\notin A$ 时 $g(x,y)$ 作为 y 的一元函数要不满足 B 中的条件。

如果 A 是递归可枚举集，事情就比较好办，可以用 A 作为条件来定义二元函数 g，如像定理8.1.3那样。如果 A 本身并不是递归可枚举集，就不能用 A 作为条件来定义函数 g，因为那样定义出来的函数不是递归函数，不能应用 s-m-n 定理。这时就需要找到一个既与 A 有关又是递归可枚举的条件。这是证明的关键也是证明的难点。下面我们列举三个例子，希望读者通过这三个例子的对比能切实地掌握这种归约方法。

例1 $K\leqslant_m \mathrm{Tot}$。

证 定义二元函数 $g(x,y)$ 为：

$$g(x,y)=\begin{cases}1 & \text{当 } x\in K\\ \text{无定义} & \text{当 } x\notin K\end{cases}$$

由于 K 是递归可枚举集，根据定理6.2.1，$g(x,y)$ 是二元递归函数。由 s-m-n 定理，存在递归全函数 $t(x)$ 使

$$g(x,y)=\Phi_{t(x)}(y)。$$

于是

$$x\in K \Rightarrow \Phi_{t(x)}\text{是常函数} \Rightarrow t(x)\in \mathrm{Tot};$$

$$x\notin K \Rightarrow \Phi_{t(x)}\text{处处无定义} \Rightarrow t(x)\notin \mathrm{Tot}。$$

所以 $K\leqslant_m \mathrm{Tot}$。

例2 $\overline{K}\leqslant_m \mathrm{Tot}$。

证 这次我们不能将 g 定义为

$$g(x,y)=\begin{cases}1 & \text{当 } x\in\overline{K}\\ \text{无定义} & \text{当 } x\notin\overline{K}\text{。}\end{cases}$$

因为 $\overline{K}$ 不是 $r.e$ 集，这样定义出来的函数 g 不是递归函数。因此也就不能对它使用 s-m-n 定理。考虑到 $\overline{K}$ 的余集 K 是 $r.e$ 集，我们可以将条件修改为：

$$g(x,y)=\begin{cases}1 & \text{当 } \Phi_{x,y}(x)=0\\ \text{无定义} & \text{当 } \Phi_{x,y}(x)=1\text{。}\end{cases}$$

由 s-m-n 定理，存在递归全函数 $t(x)$ 使

$$g(x,y)=\Phi_{t(x)}(y)\text{。}$$

如果$x\in W_x$，即$\Phi_x(x)\downarrow$，则存在y_0使得：当$y\geqslant y_0$时，$\Phi_{x,y}(x)=1$；当 $y<y_0$时，$\Phi_{x,y}(x)=0$。也就是说，当 $x\in W_x$ 时，$\Phi_{t(x)}(y)$（作为 y 的函数）仅在 $x<y_0$时有定义，是个有穷函数，当然不是全函数。

如果 $x\notin W_x$，即 $\Phi_x(x)\uparrow$，于是对任何 y，$\Phi_{x,y}(x)=0$。也就是说，当 $x\in W_x$ 时，$\Phi_{t(x)}(y)$（作为 y 的函数）是常函数，当然是全函数。

于是得到

$$x\in\overline{K}\Leftrightarrow t(x)\in\text{Tot}\text{。}$$

例3 $\text{Inf}\leqslant_m\text{Tot}$。

证 由于 Inf 和它的余集 Fin 都不是 $r.e$ 集，所以我们不能直接用它们定义函数 $g(x,y)$，而是要另选一个 $r.e$ 的谓词来定义 $g(x,y)$，并要求当 $x\in\text{Inf}$（即 W_x 无穷）时 $g(x,y)$作为 y 的一元函数是全的，而当 $x\notin\text{Inf}$（即 W_x有穷）时 $g(x,y)$作为 y 的一元函数是不全的。为此，定义

$$g(x,y)=\begin{cases}0 & \text{如果}\exists z(z>y\wedge z\in W_x)\text{，}\\ \text{无定义} & \text{否则。}\end{cases}$$

根据推论7.2.7，谓词$\exists z(z>y\wedge z\in W_x)$是 $r.e$ 的。因而由定理7.2.1，g 是递归函数。于是由 s-m-n 定理，存在递归全函数 $t(x)$使

$$g(x,y)=\Phi_{t(x)}(y)\text{。}$$

W_x 无穷 $\Rightarrow\forall y\exists z(z>y\wedge z\in W_x)\Rightarrow\Phi_{t(x)}$是零函数

$\Rightarrow t(x)\in \text{Tot}$；

W_x 有穷 $\Rightarrow$ 存在 y_0 使当 $z>y_0$ 时 $z\notin W_x$

$\Rightarrow$ 存在 y_0 使当 $y>y_0$ 时 $\Phi_{t(x)}(y)\uparrow \Rightarrow \Phi_{t(x)}$ 有穷

$\Rightarrow t(x)\notin \text{Tot}$。

所以 $\text{Inf}\leqslant_m \text{Tot}$。;

§8.3 产生集、创造集和单纯集

如今我们已经接触到三种集合：递归集(如 $\varnothing$，N 和所有的有穷集)、非递归的递归可枚举集(如 K，K_0，K_1)和非递归可枚举的集合(如 $\overline{K}$，Fin，Tot)。由定理7.2.12我们还知道非递归的递归可枚举集中都包含有无穷的递归子集。那么非递归可枚举的集合中是不是都包含有无穷的递归可枚举子集呢？我们现在就来讨论这个问题。

定义8.3.1　设 A 是自然数集，如果存在递归全函数 f，使得

只要 $W_x\subseteq A$，就有 $f(x)\in A-W_x$，

就称 A 为**产生集**，f 称为 A 的**产生函数**。

由定义容易看出产生集都不是 $r.e$ 集，因为如果 A 是 $r.e$ 集，则对某个 x，$A=W_x$，从而由 $W_x\subseteq A$ 得 $f(x)\in A-W_x$，$A-W_x\neq\varnothing$，与 $A=W_x$ 矛盾。

产生函数 f 的作用就是产生出 A 不是 $r.e$ 集的全部证据。

例1　$\overline{K}$ 是产生集。

证　由 K 的定义知

$x\in W_x\Leftrightarrow x\in K$。

因此，如果 $W_x\subseteq\overline{K}$，则 $x\notin W_x$，$x\notin K$，从而 $x\in\overline{K}$，即 $x\in\overline{K}-W_x$。这表明恒同函数 $f(x)=x$ 是 $\overline{K}$ 的产生函数。

定理8.3.1　如果 $A\leqslant_m B$ 且 A 是产生集，则 B 也是产生集。

证　设 g 是 A 的产生函数，h 是从 A 到 B 的 m-归约函数。考虑函数 $\Phi_x(h(y))$，这是一个二元的递归函数。于是由 s-m-n 定理，

存在递归全函数 $t(x)$ 使

$$\Phi_x(h(y)) = \Phi_{t(x)}(y)。\qquad ①$$

我们来证明 $h(g(t(x)))$ 是 B 的产生函数。

由①可知

$$h(y) \in W_x \text{当且仅当 } y \in W_{t(x)},$$

于是

$$h[W_{t(x)}] = W_x \text{ 并且 } W_{t(x)} = h^{-1}[W_x]^{①}。$$

如果 $W_x \subseteq B$，则

$$h^{-1}[W_x] \subseteq h^{-1}[B],$$

由于 h 是从 A 到 B 的 m-归约函数，所以 $h^{-1}[B] = A$。从而

$$W_{t(x)} = h^{-1}[W_x] \subseteq h^{-1}[B] = A。\qquad ②$$

由于 g 是 A 的产生函数，所以由②可知

$$g(t(x)) \in A - W_{t(x)}。$$

既然 $A \leqslant_m B$，于是就有

$$h(g(t(x))) \in B。\qquad ③$$

另一方面，我们已经知道 $g(t(x)) \notin W_{t(x)}$ 以及 $h[W_{t(x)}] = W_x$，所以

$$h(g(t(x))) \notin W_x。\qquad ④$$

综合③和④就得到

$$h(g(t(x))) \in B - W_x,$$

就是说 B 是产生集。

推论8.3.2　如果 $\overline{K} \leqslant_m A$，则 A 是产生集。

证　由例1和定理8.3.1立得。

利用例1和定理8.3.1我们可以得到许多产生集，比如我们已经证明 $\overline{K} \leqslant_m \text{Tot}$，因此由例1和定理8.3.1知 Tot 是产生集。推论8.3.2的逆也成立，即：

定理8.3.3　如果 A 是产生集，则 $\overline{K} \leqslant_m A$。

---

① $h^{-1}[C] = \{x \mid h(x) \in C\}$

证 设 A 是产生集，p 是其产生函数。

令

$$g(x,y,z)=\begin{cases}1 & \text{当 } y\in K \text{ 且 } z=p(x),\\ \text{无定义} & \text{否则。}\end{cases}$$

由 s-m-n 定理知存在递归全函数 $f(x,y)$ 使

$$\Phi_{f(x,y)}(z)=g(x,y,z),$$

从而

$$W_{f(x,y)}=\begin{cases}\{p(x)\} & \text{当 } y\in K\\ \varnothing & \text{当 } y\in \overline{K}。\end{cases} \quad ①$$

对 $f(x,y)$ 使用带参数的递归定理，得到递归全函数 $n(y)$，使

$$W_{n(y)}=W_{f((n(y)),y)}。 \quad ②$$

结合①和②可得

$$W_{n(y)}=W_{f(n(y)),y)}=\begin{cases}\{p(n(y))\} & \text{当 } y\in K,\\ \varnothing & \text{当 } y\in \overline{K}。\end{cases}$$

注意到 p 是产生函数，因此如果 $W_{n(y)}\subseteq A$ 则 $p(n(y))\in A-W_{n(y)}$。所以，如果 $p(n(y))\in W_{n(y)}$，则 $W_{n(y)}\not\subseteq A$。于是得到

$$y\in K \Rightarrow W_{n(y)}=\{p(n(y))\} \Rightarrow W_{n(y)}\not\subseteq A \Rightarrow p(n(y))\in \overline{A},$$

$$y\in \overline{K}\Rightarrow W_{n(y)}=\varnothing \Rightarrow W_{n(y)}\subseteq A \Rightarrow p(n(y))\in A。$$

$p(n(y))$ 当然是递归全函数，所以

$$\overline{K}\leqslant_m A。$$

推论8.3.2和定理8.3.3为我们提供了一个普遍可行的方法来证明一个集合是产生集：要证明一个集合 A 是产生集，只要证明 $\overline{K}\leqslant_m A$ 就可以了。比如我们已经证明 $\overline{K}\leqslant_m \mathrm{Tot}$，因此Tot是产生集。

由定义我们已经知道产生集都不是递归可枚举集，下面的定理叙述了产生集与递归可枚举集的另一个方面的关系：

定理8.3.4 每个产生集都有无穷的 $r.e$ 子集。

证 设 A 是产生集，f 是其产生函数。定义函数

$$g(x,y)=\begin{cases}1 & \text{如果 } x\in W_x \text{ 或 } y=f(x),\\ \text{无定义} & \text{否则。}\end{cases}$$

由于 $x\in W_x$ 和 $y=f(x)$ 都是 $r.e$ 谓词，所以如上定义的函数 g 是

递归函数。由 s-m-n 定理知存在递归全函数 $t(x)$ 使

$$g(x,y)=\Phi_{t(x)}(y)。$$

从而由 $g(x,y)$ 的定义可知

$$W_{t(x)}=W_x\cup\{f(x)\mid x\in N\}。\quad ①$$

设 e_0 是使 $W_{e_0}=\varnothing$ 的最小自然数，对任意的 n 定义

$$e_{n+1}=f(e_n)。$$

于是序列

$$e_0,\ e_1,\ \cdots,\ e_n,\ \cdots$$

（作为 n 的函数）是递归全函数，且由①可知：

$$W_{e_{n+1}}=W_{e_n}\cup\{f(e_n)\mid n\in N\}\quad ②$$

因为 f 是 A 的产生函数及 $W_{e_0}=\varnothing\subseteq A$，由②用归纳法容易验证：对每个 n

1. $W_{e_n}\subseteq A$，
2. $f(e_n)\in W_{e_{n+1}}-W_{e_n}$，
3. $f(e_n)\in A$，
4. $f(e_n)$ 作为 n 的函数是1-1的。

因而集合

$$\{f(e_0),\ f(e_1),\ \cdots\}$$

是 A 的一个无穷的 $r.e$ 子集。

产生集本身都不是 $r.e$ 集，但它的余集却可能是 $r.e$ 集，而且是一类有特殊意义的 $r.e$ 集。

定义8.3.2　如果 A 是递归可枚举集并且 $\overline{A}$ 是产生集，则称 A 为 **创造集**。

由定义容易看出创造集都不是递归集[①]。

例2　K 是创造集。因为 K 是 $r.e$ 集，而且根据例1，$\overline{K}$ 是产生

---

① 可以证明，我们常用的许多形式化的数学理论（如皮亚诺算术系统PA、策梅罗集合论系统ZF等等）的定理集都是创造集，因而都是非递归的。这表明数学研究并不能归结为一个算法，而是具有创造性的。创造集也就因此而得名。

集。

定理8.3.5　A 是创造集当且仅当 A 是 m-完全集。

证　设 A 是创造集，由定义知 $\overline{A}$ 是产生集，根据定理8.3.3，有

$$\overline{K}\leqslant_m\overline{A},$$

也就是

$$K\leqslant_m A。$$

因为 A 是 $r.e$ 集，所以由推论8.1.4知 A 是 m-完全的。

反之，如果 A 是 m-完全集，则由推论8.1.4有

$$K\leqslant_m A,$$

也就是

$$\overline{K}\leqslant_m\overline{A},$$

根据推论8.3.2，$\overline{A}$ 是产生集。由于 A 是 $r.e$ 集，所以 A 是创造集。

迄今为止我们所见到的递归可枚举集只有两类：递归集和 m-完全集。那么，有没有既非递归集也不是 m-完全的 $r.e$ 集呢？答案是肯定的。但靠指标集的方法却不能定义出这种集合来：

定理8.3.6　设 A 是指标集，如果 A 是 $r.e$ 集但不是递归集，则 A 是 m-完全的。

证　设

$$A=\{x\mid\Phi_x\in\mathscr{A}\}。$$

由于 A 不是递归集，所以 A 和 $\overline{A}$ 都不是空集。

如果空函数 $f_\varnothing\notin\mathscr{A}$，则根据定理8.2.1的证明，有

$$K\leqslant_m A,$$

因此 A 是 m-完全的。

如果 $f_\varnothing\in\mathscr{A}$，则有

$$K\leqslant_m\overline{A},$$

也就是

$$\overline{K}\leqslant_m A,$$

根据推论8.3.2，A 是产生集，从而不是 $r.e$ 集。

因此如果 A 是 $r.e$ 集但不是递归集，则 A 是 m-完全的。

定义8.3.3　如果 A 是 $r.e$ 集，$\overline{A}$ 是无穷的但 $\overline{A}$ 中不包含无穷的 $r.e$ 子集，则称 A 为**单纯集**。

由定义容易看出如果 A 是单纯集，则 A 不是递归集也不是 m-完全集。因为 $\overline{A}$ 是无穷的但 $\overline{A}$ 中不包含无穷的 $r.e$ 子集，就表明 $\overline{A}$ 本身不是 $r.e$，所以 A 就不会是递归集；由定理8.3.3，$\overline{A}$ 中不包含无穷的 $r.e$ 子集就意味着 $\overline{A}$ 不是产生集，于是 A 也就不是创造集，从而由定理8.3.5可知 A 不是 m-完全集。

定理8.3.7　*存在单纯集。*

证　定义一元函数 $f(x)$ 如下：对任何 x，依次计算

$$\Phi_x(0),\ \Phi_x(1),\cdots,\Phi_x(z),\ \cdots$$

如果找到一个 z 使 $\Phi_x(z)\downarrow>2x$，则令

$$f(x)=\Phi_x(z);$$

否则 $f(x)$ 无定义。

根据丘奇论题，f 是递归函数。我们来证明 $A=\mathrm{ran}\ f$ 是一个单纯集。

首先，$\overline{A}$ 是无穷的。因为由 f 的定义容易看出 $f(x)>2x$，于是对任何 n，序列

$$0,\ 1,\ \cdots,\ 2n$$

中至多有 n 个在 $A=\mathrm{ran}\ f$ 中，也就是说至少有 n 个在 $\overline{A}$ 中。由 n 的任意性知 $\overline{A}$ 是无穷的。

其次，$\overline{A}$ 中不包含无穷的 $r.e$ 子集。如果 B 是无穷的 $r.e$ 集，则对某个 e，$B=E_e$。由于 E_e 无穷，故存在 z 使

$$\Phi_e(z)>2e,$$

则由 f 的定义知对某个这样的 z

$$f(e)=\Phi_e(z)\in A\cap E_e。$$

也就是说 $E_e\not\subseteq\overline{A}$。

第三，A 是 $r.e$ 集，因为它是递归函数的值域。

我们已经看到单纯集的余集不是 $r.e$ 集，也不包含无穷的 $r.e$

子集。因此，如果 A 是单纯集，$W_x \subseteq A$，则 W_x 是有穷的。

定义8.3.4　设 A 是单纯集，如果 A 满足：存在递归全函数 $g(x)$ 使得

$$\text{如果 } W_x \subseteq A, \text{ 则 } |W_x| < g(x),$$

则称 A 为**能行单纯集**。

能行单纯集的特点就是可以由一个递归全函数给出该集合中(有穷的)$r.e$ 子集的基数的上界。定理8.3.6的证明中所定义的那个单纯集就是一个能行单纯集。因为在证明过程中我们已经看到

$$\text{如果 } E_e \subseteq A, \text{则 } |E_e| < 2e+1。$$

根据第七章习题5，存在递归全函数 $h(x)$ 使 $W_x = E_{h(x)}$，于是

$$\text{如果 } W_x \subseteq A, \text{则 } |W_x| = |E_{h(x)}| < 2h(x)+1,$$

所以 A 是能行单纯集。

习　　题

1. 证明以下集合是产生集：

(1) $\{x | W_x \text{ 有穷}\}$，

(2) $\{x | E_x \neq N\}$，

(3) $\{x | \Phi_x \text{ 是1－1的}\}$。

2. 证明以下集合是创造集：

(1) $\{x | x \in E_x\}$，

(2) $\{x | \Phi_x(x) \in A\}$(其中 A 是一个任意的非空 $r.e$ 集)，

(3) $\{x | \Phi_x(x) = f(x)\}$(其中 f 是一个固定的递归全函数)。

3. 证明：如果 A 是 $r.e$ 集，$A \cap B$ 是产生集，则 B 是产生集。

4. 证明：如果 C 是创造集，A 是 $r.e$ 集，$A \cap C = \varnothing$，则 $C \cup A$ 是创造集。

第九章

图灵度

m-度在一定程度上刻划了函数(集合、谓词)计算(判定)问题的难度,具有相同的m-度的函数(集合、谓词)的计算(判定)问题的难度可以认为是相同的:如果 $f\equiv_m g$,那么只要我们会计算 f 也就会计算 g;只要会计算 g 也就会计算 f。

但是,m-度并没有完全刻画出计算(判定)难度之间的这种关系,例如我们容易想象,对任何的集合 A, A 的判定问题与 $\overline{A}$的判定问题应当是有相同的难度的:只要会判定 A,自然也就会判定 $\overline{A}$;只要会判定 $\overline{A}$ 自然也就会判定 A。然而并不是对任何集合 A 都有 $A\equiv_m\overline{A}$ 的,例如我们已经知道 $K\not\equiv_m\overline{K}$,实际上是既没有 $K\leqslant_m\overline{K}$ 也没有 $\overline{K}\leqslant_m K$。也就是说 K 和 $\overline{K}$ 在$\leqslant_m$ 下是不可比较的。为了更精确地刻画函数(集合、谓词)之间计算(判定)问题的难易差别,我们在这一章中讨论相对可计算性和相对递归性概念。

§9.1 相对可计算性

常用的刻画相对可计算性的方法是带神谕的图灵机和相对递归函数。

带神谕的图灵机 是在普通的图灵机之上增加一条可以向两端无限伸长的带子,称为**神谕带**。其结构如图9.1所示。神谕带上也被分成大小相等的方格,神谕带与工作带的方格对齐,读写头在两条带子中间。读写头可以读出工作带上的字母,可以在工作带上写

下字母；读写头可以读出神谕带上的字母，但不能在神谕带上写字母；读写头可以沿着两条带子左右移动(每次移动一格)。神谕带上有一个方格标有“原点”的标志。神谕带上装有一个集合 A 的特征函数，具体做法是：从原点开始依次向右在各方格中写有

$$C_A(0),\ C_A(1),\ C_A(2),\cdots$$

原点左边的方格都是空格，用 b 表示。就是说神谕带上可以有三种字母：b,0和1。如果集合 A 不是递归集(我们感兴趣的正是这种情形)，向神谕带上写 A 的特征函数这件事就不是机器和人所能完成的，得去请教“神”，由“神”来回答 $C_A(x)$ 究竟是0还是1，所以这条带子叫神谕带，带上所装的集合 A 称为**神谕**。

今后，我们将集合 A 和它的特征函数 C_A 等同看待，就用$A(x)$表示 $C_A(x)$，$A(x)=0$就意味着 $x\notin A$，$A(x)=1$就意味着$x\in A$。

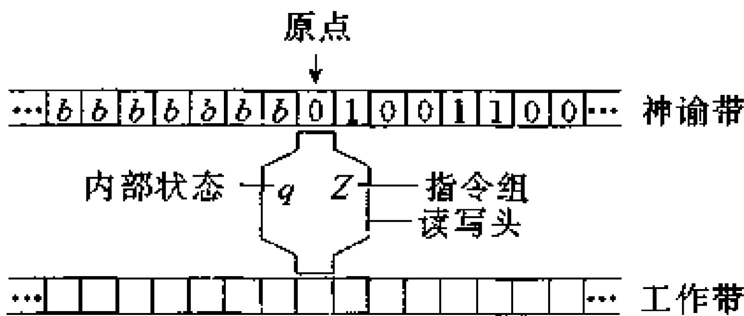


图9.1　带神谕的图灵机

工作带上的字母表的情形仍与普通图灵机一样。读写头仍可以处于不同的内部状态，读写头仍是在指令的指挥下进行运算，不过这种带神谕的图灵机的指令不再是四元组，而是五元组。指令有三种形式

$$q_i\, S_j\, S_k\, S_m\, q_l,$$

$$q_i\, S_j\, S_k\, R\, q_l,$$

$$q_i\, S_j\, S_k\, L\, q_l。$$

其中的 S_j 表示读写头在工作带上读到的字母，S_k 表示读写头在神谕带上读到的字母。$q_i\, S_j\, S_k\, S_m\, q_l$ 的意思是读写头在工作带上读到 S_j、在神谕带上读到 S_k 后，将工作带上的 S_j 改为 S_m，读写

头位置不动，状态变为 q_l。

$q_i\ S_j\ S_k\ R\ q_l$ 的意思是读写头在工作带上读到 S_j、在神谕带上读到 S_k 后，工作带上的内容不变，读写头右移一格，状态变为 q_l。

$q_i\ S_j\ S_k\ R\ q_l$ 的意思是读写头在工作带上读到 S_j、在神谕带上读到 S_k 后，工作带上的内容不变，读写头左移一格，状态变为 q_l。

在本章和下一章中我们所谈的图灵机都是带神谕的图灵机，因此下面我们将省略“带神谕的”这个定语，简称图灵机。和第二章的情形相似，这部图灵机和那部图灵机的区别就在于它们的指令组的不同，因此我们仍是把图灵机与其指令组等同看待。

输入的方法基本上与普通图灵机的情形相同，只是要求输入时最左边的1对准神谕带的原点，输入时读写头的状态为 q_0，位置对准原点。运算、停机、计算、输出等概念都可以根据普通图灵机的情形照搬过来，这里就不重复了。需要注意的是：带神谕的图灵机的运算、计算不仅与图灵机的指令组和输入有关，而且与神谕带上的集合 A 有关。这一点对于理解下一章的内容至关重要，我们来看两个例子。

例1　设图灵机 $Z_1=\{q_0\ 0\ 1\ R\ q_0,\ q_0\ 1\ 1\ R\ q_0\}$。$Z_1$的作用是：只要在神谕带上读到1（不管在工作带是读到什么信息），读写头就右移一格，状态不变，直到在神谕带上找到一个0为止。因此如果神谕 $A\neq N$，则无论输入什么，Z_1总能停机，而且工作带上内容不变；但如果神谕 $A=N$，则无论输入什么，Z_1都不能停机。

例2　令 Z_2由如下指令组成：

$q_0\ 0\ 0\ R\ q_2$,

$q_0\ 1\ 0\ R\ q_2$,

$q_0\ 0\ 1\ R\ q_1$,

$q_0\ 1\ 1\ R\ q_1$,

$q_1\ 0\ 1\ R\ q_0$,

$q_1\ 1\ 1\ R\ q_0$,

$q_2\ 0\ 0\ R\ q_2$,

$q_2\ 1\ 0\ R\ q_2$,

$q_2\ 0\ 1\ R\ q_2$,

$q_2\ 1\ 1\ R\ q_2$。

Z_2的运行情况是：如果神谕带上没有0或者神谕带上的第一个0是在偶数位置，读写头一味右移，不停机；如果神谕带上的第一个0是在奇数位置，读写头移到这个位置上停机。

当神谕 A 确定之后，(带神谕的)图灵机 Z 的输入与输出之间的关系就是一个部分函数，称为图灵机 Z 在神谕 A 下所计算的 n 元函数，记作

$$\Phi_Z^{A,(n)}(x_1,\cdots,x_n)。$$

定义9.1.1　设 $A\subseteq N$ 是自然数集，$f(x_1,\cdots,x_n)$是 n 元函数，如果存在图灵机 Z 使 $f=\Phi_Z^A$，则称 f 为相对于 A-可计算的函数，简称 **A-可计算函数**。

定义9.1.2　从初始函数(零函数、后继函数、投影函数)和集合 A(的特征函数)出发，经有穷次使用复合、原始递归和取极小运算所得到的函数称为相对于 A-递归的(部分)函数，简称 **A-递归函数**。如果一个 A-递归函数的定义域是全的，就称为 **A-递归全函数**。

除了集合的特征函数之外，我们也可以对任意的部分函数 f 定义 f-递归函数：将 f 加入初始函数就可以了。利用与第二、第四章类似的方法，可以证明：

定理9.1.1　函数 f 是 A-递归的，当且仅当 f 是 A-可计算的。

就是说，定义9.1.1和定义9.1.2所定义的函数类是相同的。今后在讨论 A-递归函数的性质时，我们有时使用定义9.1.1比较方便，有时使用定义9.1.2比较方便。

例如，由定义9.1.2容易看出对任何集合 A，$\overline{A}$ 是 A-递归的，因为

$$\overline{A}(x)=1-A(x)。$$

但如果要用定义9.1.1去证明 $\overline{A}$ 是 A-可计算的，就得构造在神谕

A 下可以计算 $\overline{A}$ 的图灵机，过程就要烦琐一些（见习题1）。因此从现在起我们不再区分 A-可计算函数和 A-递归函数这两个概念而是笼统地称为 A-递归函数。

我们仍可以像第四章那样给（带神谕的）图灵机编号，使得每部图灵机 Z 有一个号码 $\#(Z)$，而每个自然数都表示唯一的一部图灵机。如果 $\#(Z)=e$，我们就将 $\Phi_Z^{A,(n)}(x_1,\cdots,x_n)$ 记作

$$\Phi_e^{A,(n)}(x_1,\cdots,x_n) \quad \text{或} \quad \{e\}^{A,(n)}(x_1,\cdots x_n)$$

因此全体 n 元的 A-递归函数（A-可计算函数）都包括在序列

$$\Phi_0^{A,(n)},\ \Phi_1^{A,(n)},\ \cdots,\ \Phi_e^{A,(n)},\ \cdots \qquad ①$$

之中，而且每个 n 元 A-递归函数在序列①中出现无穷多次。序列①也记作

$$\{0\}^{A,(n)},\ \{1\}^{A,(n)},\ \cdots,\ \{e\}^{A,(n)},\ \cdots \qquad ②$$

当元数 $n=1$时，或根据上下文 n 的数值，省略之后不致导致歧义时，序列①，②中的上标(n)可以省略，比如说，当 $n=1$时，序列①②可以分别写成

$$\Phi_0^A,\ \Phi_1^A,\ \cdots,\Phi_e^A,\ \cdots \qquad ①'$$

和

$$\{0\}^A,\ \{1\}^A,\ \cdots,\ \{e\}^A,\ \cdots \qquad ②'$$

每个一元 A-递归函数都在序列①′，②′中出现无穷多次。当神谕 A 是空集$\varnothing$时也可以略去不写，就是说可以将

$\{e\}^{\varnothing}(x)$简写为$\{e\}(x)$。

利用定义9.1.2和第三章的内容，容易证明

定理9.1.2　设 f 是部分函数，A 是自然数的集合，若 A 是递归集，则

f 是 A-递归函数 当且仅当 f 是递归函数；

特别地

f 是$\varnothing$递归函数 当且仅当 f 是递归函数。

定理9.1.3　如果 f 是递归函数，则对任何集合 A，f 是 A-递归函数。

由于配对函数是递归全函数，所以对任何集合 A，配对函数都是 A-递归的，因此我们可以只讨论一元的 A-递归函数。多元的情形都可以通过配对函数化归为一元的情形。作为特例，序列

$$\{0\},\ \{1\},\ \{2\},\ \cdots,\ \{e\},\ \cdots$$

中包括了全体一元的递归函数，而且每个递归函数在其中出现无穷多次。

此外，第二——第五章中全部有关递归函数的内容，都可以用同样的方法移植到相对递归函数上来，例如我们可以证明

定理9.1.4（**相对的 s-m-n 定理**） 对任意的 $m+n$ 元 A-递归函数 $f(x_1,\cdots,x_n,y_1,\cdots,y_m)$，存在 m 元递归全函数 $t(y_1,\cdots,y_m)$，使

$$f(x_1,\cdots,x_n,y_1,\cdots,y_m)=\Phi^A_{t(y_1,\cdots,y_m)}(x_1,\cdots,x_n)。$$

定理9.1.5（**相对的递归定理**） 对任何 A-递归全函数 $f(x)$，存在自然数 n，使得

$$\Phi^A_n=\Phi^A_{f(n)} \quad 以及 \quad W^A_n=W^A_{f(n)}。$$

定理9.1.6（**相对的带参数的递归定理**） 对任何 A-递归全函数 $f(x,y)$，存在递归全函数 $n(y)$，使得

$$\Phi^A_{n(y)}=\Phi^A_{f(n(y),y)} \quad 以及 \quad W^A_{n(y)}=W^A_{f(n(y),y)}。$$

注意：定理中的 $n(y)$ 是递归函数，与神谕 A 无关。

今后，在证明中我们也使用相对的丘奇论题，就是说：要证明一个函数 f 是 A-递归的，只要给出一个算法使得我们能够在知道对任意的自然数 x 是否有 $x\in A$ 的前提下计算 f 的值，就可以了。

当我们使用第 e 号图灵机在神谕 A 下输入自然数 x 进行运算时，如果 $\{e\}^A(x)\downarrow$，则运算将在有穷步内结束，形成一个计算。由于计算是有穷的，因此在整个计算过程中读写头只会读到神谕带上的有穷多个格中的数据（0或1）。如果在计算 $\{e\}^A(x)$ 的过程中，读写头所到达的最右边的位置是从原点算起的第 n 个方格，我们就说这次计算所用到的**神谕数**是 n。这就是说，对于任何 $m>n$，$A(m)$ 的值对这个计算的过程和结果都没有任何影响，因为读写头

移不到写有 $A(m)$的位置。

以下，我们用小写的希腊字母 σ,τ 表示有穷长的0,1序列，用 $\sigma(x)$ 表示序列 σ 中的第 x 个元素(0或1)。换句话说，σ,τ 也就表示在自然数的一个前段$\{0,1,2,\cdots,k\}$上定义的0,1值有穷函数。今后，我们用 $lh(\sigma)$表示序列的长度，即

$$lh(\sigma)=\mu x[\sigma(x)\uparrow]。$$

由于我们可以在有穷的0,1序列和自然数之间建立能行的一一对应，因此可以把有穷的0,1序列与自然数等同看待，我们可以在 σ,τ 前加量词，如$\forall\sigma\subset A$，$\exists\tau\supset\sigma$ 等等。这里的 σ,τ 就具有双重身份——它们既代表一个自然数，又代表一个有穷的0,1序列。一个集合(无论是有穷集还是无穷集)A 的特征函数可以看成是一个无穷的0,1序列，其中第 x 个元素就是 $A(x)$(我们已经将集合与它的特征函数等同看待)的值。今后，我们用 $A\upharpoonright x$ 表示将无穷的0,1序列限制在小于 x 的自然数前段上所得到的有穷0,1序列，即

$$(A\upharpoonright x)(y)=\begin{cases}A(y) & \text{如果 } y<x,\\ \uparrow & \text{如果 } y\geqslant x\end{cases}$$

从而对任何一个有穷的0,1序列 σ，有无穷多个集合(无穷的0,1序列)A 使得 $\sigma\subset A$；对任何一个集合 A，也有无穷多个(长度彼此不同的)σ 使 $\sigma\subset A$。

今后，我们还要经常使用以下记号和概念：

定义9.1.3

(1) 等式$\{e\}_s^A(x)=y$ 意思是

$0\leqslant x,y,e<s$；

并且

在神谕 A 下向第 e 号图灵机输入 x 后，在 s 步之内停机，输出为 y；

并且

在计算$\{e\}^A(x)$的过程中所用到的神谕数 $n<s$。

如果存在 y 使$\{e\}_s^A(x)=y$，就记作$\{e\}_s^A(x)\downarrow$。

(2) **使用函数** $u(A;e,x,s)$定义为：

$$u(A;e,x,s)=\begin{cases}1+n & \text{如果}\{e\}_s^A(x)\downarrow\text{且该计算所用到的神谕数为 } n, \\ 0 & \text{否则。}\end{cases}$$

使用函数 $u(A;e,x)$定义为

$$u(A;e,x)=\begin{cases}u(A;e,x,s) & \text{如果存在 } s \text{ 使}\{e\}_s^A(x)\downarrow \\ \uparrow & \text{否则。}\end{cases}$$

(3) 等式$\{e\}_s^\sigma(x)=y$的意思是

对某个$A\supset\sigma$，$\{e\}_s^A(x)=y$

并且

$u(A;e,x)<lh(\sigma)$。

(4) 等式$\{e\}^\sigma(x)=y$的意思是

存在 s 使$\{e\}_s^\sigma(x)=y$

在定义9.1.3中，计算步数 s 有着重要的上界作用，这表现在两个方面：

1. 如果$\{e\}_s^A(x)=y$，则 $e,x,y<s$ 并且 $u(A;e,x,s)\leqslant s$；
2. 如果$\{e\}_s^A(x)=y$，则对任何 $t\geqslant s$

$\{e\}_t^A(x)=y$ 并且 $u(A;e,x,s)=u(A;e,x,t)$。

注意：定义9.1.3之(3)(4)中所定义的使用函数与 A 有关。由丘奇论题容易看出根据上述定义，我们可以得到以下一些重要性质：

定理9.1.7

(1) 集合

$\{(e,\sigma,x,s)\mid\{e\}_s^\sigma(x)=y\}$

是递归集；

(2) 集合

$\{(e,\sigma,x)\mid\{e\}^\sigma(x)=y\}$

是 $r.e$ 集。

定理9.1.8

(1) 如果$\{e\}^A(x)=y$，则存在s和$\sigma\subset A$，使$\{e\}_s^\sigma(x)=y$；

(2) 如果$\{e\}_s^\sigma(x)=y$,则对任何$t\geqslant s$和任何$\tau\supseteq\sigma$，

$\{e\}_t^\tau(x)=y$；

(3) 如果$\{e\}^\sigma(x)=y$,则对任何$A\supset\sigma$有$\{e\}^A(x)=y$。

仿照定义7.2.1,我们有

定义9.1.5　A-递归函数的定义域称为**A-递归可枚举集**(A-$r.e$ **集**)。A-递归函数$\Phi_e^{A,(n)}$的定义域记作

$$W_e^{A,(n)}=\mathrm{dom}\ \Phi_e^{A,(n)}。$$

当$n=1$时，上标中的(n)可以省略。于是全体一元的A-递归可枚举集都包含在序列

$$W_0^A, W_1^A, \cdots, W_e^A, \cdots$$

中,而且每个出现无穷多次。

当神谕A为空集$\varnothing$时,上标$\varnothing$可以略去不写,即将$W_e^\varnothing$简写为W_e。事实上，序列

$$W_0^\varnothing, W_1^\varnothing, \cdots, W_e^\varnothing, \cdots$$

就是全体$r.e$集(每个$r.e$集在其中出现无穷多次)。

将第七章的定理7.2.1、定理7.2.3相对化,我们有

定理9.1.9　集合B是A-递归可枚举的当且仅当函数

$$f(x)=\begin{cases}1 & \text{如果 } x\in B\\ \uparrow & \text{如果 } x\notin B\end{cases}$$

是A-递归函数。

定理9.1.10　集合B是A-递归的当且仅当B和$\overline{B}$都是A-递归可枚举集。

定理9.1.11　以下三个条件彼此等价：

(1) B是A-递归可枚举集；

(2) $B=\varnothing$或B是某个A-递归全函数的值域；

(3) 存在A-递归谓词$R(x,y_1,\cdots,y_m)$使

$$x\in B\Leftrightarrow\exists y_1\cdots\exists y_m R(x,y_1,\cdots,y_m)。$$

定理9.1.11是定理7.2.9的相对化形式，其证明与定理7.2.9相似，此处从略。

今后我们也常要用到如下的记号：

定义9.1.6　定义 $W^A_{e,s}$和 $W^\sigma_{e,s}$为

$$W^A_{e,s}=\{x\mid\exists y(\{e\}^A_s(x)=y)\},$$

$$W^\sigma_{e,s}=\{x\mid\exists y(\{e\}^\sigma_s(x)=y)\},$$

对照定义9.1.3，容易得到以下性质：

定理9.1.12

(1) $W^A_{e,s}\subseteq W^A_e$；

(2) $x\in W^A_{e,s}\Rightarrow x<s$，从而$|W^A_{e,s}|\leqslant s$；

(3) 对任何 $A\supset\sigma$，$W^\sigma_{e,s}\subseteq W^A_e$；

(4) $x\in W^\sigma_{e,s}\Rightarrow x<s$，从而$|W^\sigma_{e,s}|\leqslant s$。

§9.2　图　灵　度

利用相对可计算性给自然数分类，所得到的就是图灵度的概念。

定义9.2.1　设 A，B 是集合，如果 B 是 A-递归的，就称 B 可以**图灵归约**到 A，记作

$$B\leqslant_T A;$$

如果 $A\leqslant_T B$ 且 $B\leqslant_T A$，则称 A 和 B **图灵等价**，记作

$$A\equiv_T B。$$

由定义容易验证，$\leqslant_T$ 是偏序关系，即满足对任意的自然数集 A，B，C

1. $A\leqslant_T A$(自反性)；

2. 如果 $A\leqslant_T B$ 且 $B\leqslant_T C$，则 $A\leqslant_T C$(传递性)。

$\equiv_T$ 是等价关系，即满足对任意的自然数集 A，B，C

3. $A\equiv_T A$(自反性)；

4. 如果 $A\equiv_T B$ 且 $B\equiv_T C$，则 $A\equiv_T C$(传递性)；

5. 如果 $A\equiv_T B$，则 $B\equiv_T A$（对称性）。

由定义9.2.1和定义8.1.1可以得到

定理9.2.1　设 A,B 是集合。

(1) 如果 $A\leqslant_m B$，则 $A\leqslant_T B$；

(2) 如果 $A\equiv_m B$，则 $A\equiv_T B$。

证　设 $A\leqslant_m B$，则由定义8.1.1知存在递归全函数 f，使

$$x\in A \Leftrightarrow f(x)\in B。$$

B 当然是 B-递归的，f 是递归函数，因而也是 B-递归的，因此

$$f(x)\in B$$

是 B-递归的谓词，所以 A 是 B-递归集，即 $A\leqslant_T B$。由(1)自然就得到(2)。

定理9.2.1的逆并不成立，因为 $K\equiv_T\overline{K}$，但我们知道既没有 $K\leqslant_m\overline{K}$，也没有 $\overline{K}\leqslant_m K$。

定义9.2.2　自然数集在 $\equiv_T$ 关系下的等价类称为**图灵度**，简称**度**。含有集合 A 的图灵度记作 deg(A)，即

$$\deg(A)=\{B\mid B\equiv_T A\}。$$

如果 B 是 $r.e$ 集，则称 deg(B)为**递归可枚举的度**（$r.e$ **度**）；如果 B 是 A-递归可枚举集（A-$r.e$ 集）则称 deg(B)是 A-递归可枚举（A-$r.e$）的度。

由于 $\equiv_T$ 是等价关系，所以当 $A\equiv_T B$ 时 deg(A)＝deg(B)。另外，需要注意图灵度和 m-度之间的一个明显区别：递归可枚举的 m-度中的集合都是递归可枚举集，而递归可枚举的图灵度中可能有非递归可枚举集。因为对任何递归可枚举集 A，$\overline{A}\in\deg(A)$，而我们已经知道递归可枚举集的余集未必是递归可枚举集。

今后我们也使用黑正体的小写拉丁字母 **a**，**b**，**c**，…表示图灵度：

$$\mathbf{a}=\deg(A),$$

$$\mathbf{b}=\deg(B),$$

$$\mathbf{c}=\deg(C),$$

等等。

全体图灵度所组成的集合记作 **D**。

定义9.2.3　设 A，B 是自然数集，图灵度 $\mathbf{a}=\deg(A)$，$\mathbf{b}=\deg(B)$，如果 $A\leqslant_T B$，就记 $\mathbf{a}\leqslant\mathbf{b}$；如果 $\mathbf{A}\leqslant_T\mathbf{B}$ 且 $\mathbf{B}\not\leqslant_T \mathbf{A}$，则记 $\mathbf{a}<\mathbf{b}$。

容易验证，如上定义的 **D** 上的$\leqslant$关系是偏序关系（自反、传递），**D** 上的$<$关系是严格偏序。从定义中我们看不出图灵度之间是否具有可比较性（即对任何两个图灵度 $\mathbf{a}$ 和 $\mathbf{b}$，或者 $\mathbf{a}\leqslant\mathbf{b}$ 或者 $\mathbf{b}\leqslant\mathbf{a}$），到下一章我们会证明图灵度之间并不普遍具有可比较性，也就是说存在两个图灵度 $\mathbf{a}$ 和 $\mathbf{b}$ 使得既没有 $\mathbf{a}\leqslant\mathbf{b}$ 也没有 $\mathbf{b}\leqslant\mathbf{a}$。

定义9.2.4

(1) 集合

$$K^A=\{x\mid\{x\}^A(x)\downarrow\}$$

称为 A 的**跳跃集**，也记作 A'。算子 “$'$” 称为**跳跃算子**

(2) 归纳定义 A 的 n 级跳跃 $A^{(n)}$ 为：

$$A^{(0)}=A,\quad A^{(k+1)}=(A^{(k)})'。$$

将§7.2中的内容相对化，我们可以定义

$$K_0^A=\{(x,y)\mid y\in W_x^A\},$$

$$K_1^A=\{x\mid W_x^A\neq\varnothing\}。$$

根据§8.1的例1和例2，我们知道

$$K\equiv_m K_0\equiv_m K_1, \qquad ①$$

于是由定理9.2.1知

$$K\equiv_T K_0\equiv_T K_1。\qquad ②$$

将①的证明相对化，可以得到

$$K^A\equiv_m K_0^A\equiv_m K_1^A,$$

从而由定理9.2.1知

$$K^A\equiv_T K_0^A\equiv_T K_1^A。$$

对于跳跃算子，我们有如下的重要定理：

定理9.2.2(**跳跃定理**)

(1) A'是A-$r.e$集；

(2) $A' \not\leqslant_T A$；

(3) B是A-$r.e$集 当且仅当 $B \leqslant_m A'$；

(4) 如果A是B-$r.e$集且$B \leqslant_T C$,则A是C-$r.e$集；

(5) $B \leqslant_T A$ 当且仅当 $B' \leqslant_m A'$；

(6) $B \equiv_T A$ 当且仅当 $B' \equiv_m A'$。

证 (1)—(3)分别是§7.2之例3、定理7.1.1和定理8.1.3的相对化形式，请读者参照§7.2之例3、定理7.1.1和定理8.1.3的证明写出它们的证明。下面证明(4)—(6)。

对于(4),如果$A=\varnothing$，A是C-递归集,当然是C-$r.e$集。如果$A \neq \varnothing$,则根据定理9.1.11，A是某个B-递归全函数f的值域。由于B是C-递归集,所以f也是C-递归全函数。因此A是一个C-递归全函数的值域,也就是说A是C-$r.e$集。

对于(5),设$B \leqslant_T A$。由(1)知B'是B-$r.e$集,根据(4)，B'也是A-$r.e$集。再由(3)即得到$B' \leqslant_m A'$。

反之,设$B' \leqslant_m A'$,由定义9.1.2可知B和$\overline{B}$都是B-递归的,因而也就都是B-$r.e$的,由(3)可得$B \leqslant_m B'$及$\overline{B} \leqslant_m B'$。而$B' \leqslant_m A'$,于是有$B \leqslant_m A'$及$\overline{B} \leqslant_m A'$。再由(3)(的另一个方向)即得到$B$和$\overline{B}$都是$A$-$r.e$集,根据定理9.1.10即有$B \leqslant_T A$。

(6)是(5)的直接推论。

跳跃定理为我们提供了有关图灵度的许多信息。比如,我们已经知道对任何的集合A，$A \leqslant_T A'$,因此$\deg(A) \leqslant \deg(A')$。现在我们又知道$A' \not\leqslant_T A$(定理9.2.2之(2)),于是就得到对任意的集合A

$$\deg(A) < \deg(A')。$$

今后,当$\mathbf{a}=\deg(A)$时,我们就记$\mathbf{a}'=\deg(A')$。从而对任何图灵度$\mathbf{a}$,有$\mathbf{a}<\mathbf{a}'$。图灵度$\mathbf{a}'$称为图灵度$\mathbf{a}$的**跳跃**。我们用$\mathbf{0}$表示全体递归集所组成的图灵度,即

$$\mathbf{0}=\deg(\varnothing)$$

于是

$$\mathbf{0}'=\deg(\varnothing')$$

$$\mathbf{0}''=\deg(\varnothing'')$$

…………

$$\mathbf{0}^{(n)}=\deg(\varnothing^{(n)})$$

它们是彼此不同的图灵度：

$$\mathbf{0}<\mathbf{0}'<\mathbf{0}''<\cdots<\mathbf{0}^{(n)}<\cdots$$

其中$\varnothing'=K^{\varnothing}\equiv_T K\equiv_T K_0\equiv_T K_1$

此外，由定理9.1.3可知0是最小的图灵度。全体图灵度的结构如图9.2所示：

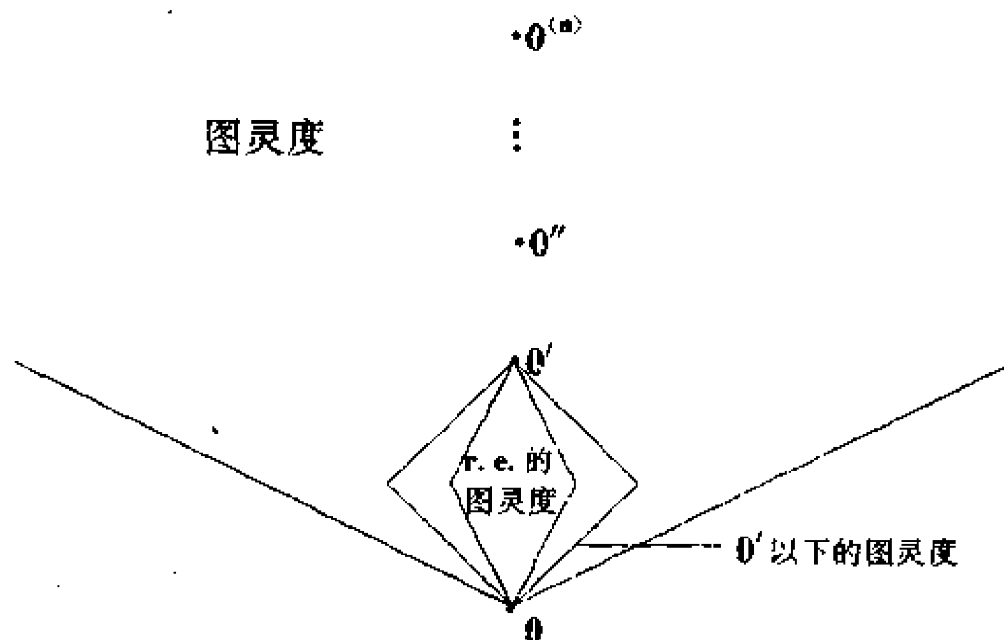


图9.2 图灵度的结构

图9.2中0与0′之间的大菱形表示大于等于0而又小于等于0′的图灵度，0与0′之间的小菱形表示全体递归可枚举的图灵度。现在我们还不知道这两个菱形的部分除了0和0′有没有其他的度。现代递归论的主要内容就是研究$r.e$的图灵度的结构，也就是要弄

清这个小菱形的内部是个什么样子。

§9.3 极　　限

我们今后的工作就是构造一些集合，使得它的图灵度具有某种性质。为此，引入全函数序列的极限的概念。

定义9.3.1　(1) 设 $f_0, f_1, \cdots, f_s, \cdots$ 是由(一元)递归全函数组成的函数序列，如果 $f_s(x)$ 作为 s 和 x 的二元函数是递归函数，则称该函数序列为**递归序列**。今后将全函数序列

$$f_0, f_1, \cdots, f_s, \cdots$$

简记作 $\{f_s\}_{s\in N}$。

(2) 设有全函数序列 $\{f_s\}_{s\in N}$ 和全函数 f，如果对每个 x，都存在(与 x 有关的) s，使得当 $t\geqslant s$ 时 $f_t(x)=f(x)$，则称函数 f 是序列 $\{f_s\}_{s\in N}$ 的**极限**，也称序列 $\{f_s\}_{s\in N}$ **收敛**于 f，记作

$$\lim_s f_s=f。$$

(3) 设 $\lim_s f_s=f$，$m(x)$ 是一个全函数，如果对每个 x, s 都有

如果 $s\geqslant m(x)$，则 $f_s(x)=f(x)$，

则称 $m(x)$ 为 f_s 的**收敛模**，简称**模**。如果模 m 满足

$$m(x)=\mu s[(\forall t\geqslant s)(f_t(x)=f(x))]$$

则称 m 为($\{f_s\}_{s\in N}$的)**最小模**。

从直观上讲，$\lim_s f_s=f$ 就是说对任何 x，数列 $\{f_s(x)\}_{s\in N}$ 中至多只有有穷多个不等于 $f(x)$。如果一个函数序列 $\{f_s\}_{s\in N}$ 的极限 f 是个0,1值函数，那么 f 就是某个集合的特征函数。因此我们可以使用函数序列来定义我们所需要的集合。

对于递归的函数序列 $\{f_s\}_{s\in N}$(在本书中只讨论递归的函数序列)，如果

$$\lim_s f_s=f$$

且 $m(x)$ 是它的一个模，则由定义容易看出

$$f\leqslant_T m。$$

因为 $f(x)=f_{m(x)}(x)$。但是却不能保证 $m\leqslant_T f$，因为我们虽然知道只有有穷多个 s 使 $f_s(x)\neq f(x)$，但并不总能知道究竟有多少个这样的 s。不过，在附加了一些条件后，也还是可以保证 $m\leqslant_T f$ 的。例如

定理9.3.1 如果递归的函数序列 $\{f_s\}_{s\in N}$ 有极限 f，且对任何 x，$f_s(x)$ 对 s 是单调上升的（即当 $t>s$ 时 $f_t(x)\geqslant f_s(x)$），则 $\{f_s\}_{s\in N}$ 的最小模 m 满足

$$m\leqslant_T f。$$

证 对于任给的 x，可以 f 递归地计算 $m(x)$：(1)先算出 $f(x)$，这个过程当然是 f 递归的。然后，(2)依次计算

$$f_0(x),\ f_1(x),\ f_2(x),\ \cdots$$

直到找到一个最小的 s_0 使 $f_{s_0}(x)=f(x)$。由 $\{f_s\}_{s\in N}$ 是递归序列而且

$$\lim_s f_s=f$$

以及我们已经算出了 $f(x)$ 的值，所以(2)的过程是递归的。这个 s_0 就是 $m(x)$ 的值，因为 $f_s(x)$ 对 s 的单调上升性保证对任何 $t\geqslant s$

$$f(x)=f_{s_0}(x)\leqslant f_t(x)\leqslant f(x)。$$

由于(1)是 f 递归的，(2)是递归的，所以整个过程是 f 递归的。根据相对的丘奇论题，就得到 $m\leqslant_T f$。

对于比较一般的情形，有

定理9.3.2（模引理） 如果 A 是递归可枚举集并且全函数 $f\leqslant_T A$，则存在递归序列 $\{f_s\}_{s\in N}$ 使

$$\lim_s f_s=f$$

并且存在一个模 m 使 $m\leqslant_T A$。

证 设 $A=W_i$，$A_s=W_{i,s}$，$f=\{e\}^A$。定义全函数

$$f_s(x)=\begin{cases}\{e\}_s^{A_s}(x) & \text{如果}\{e\}_s^{A_s}(x)\downarrow\\ 0 & \text{否则}\end{cases}$$

由于 $A_s=W_{i,s}$ 是有穷集，$|A_s|\leqslant s$，$x\in A_s$ 是关于 s,x 的二元递归谓词（i 是常数），所以 $\{e\}_s^{A_s}(x)\downarrow$ 是关于 s,x 的递归谓词（e 是常数）。

因而 $f_s(x)$ 是 s,x 的二元递归函数。由定义容易看出

$$\lim_s f_s = f。$$

令

$$m(x) = \mu s[(\exists z \leqslant s)(\{e\}_s^{A_s \upharpoonright z}(x) \downarrow \wedge A_s \upharpoonright z = A \upharpoonright z)]。$$

由 $x \in A_s$ 是(关于 x,s 的)二元递归谓词，于是 $x \in A_s \upharpoonright z$ 就是(关于 x,s,z 的)三元递归谓词，因此

$$\{e\}_s^{A_s \upharpoonright z}(x) \downarrow$$

是(关于 x,s,z 的)三元递归谓词。而

$$A_s \upharpoonright z = A \upharpoonright z$$

是(关于 s,z 的)二元 A-递归谓词，于是

$$\{e\}_s^{A_s \upharpoonright z}(x) \downarrow \wedge A_s \upharpoonright z = A \upharpoonright z \qquad ①$$

是 A-递归的。加上有界量词 $\exists z \leqslant s$ 仍是 A-递归的。所以 $m(x)$ 是 A-递归函数。

由于 f 是全函数，即对任何 x，$\{e\}^A(x) \downarrow$，因此满足①的 s 和 z 总是有的(用 z_0 表示使①为真的最小的 z，z_0 可以 A-递归地算出)，就是说 $m(x)$ 是全函数。。

最后，$m(x)$ 是模，因为当 $s \geqslant m(x)$ 时，

$$f_s(x) = \{e\}_s^{A_s \upharpoonright z}(x) = \{e\}_s^{A \upharpoonright z_0}(x) = \{e\}^A(x) = f(x)。$$

定理9.3.3(**极限引理**) 对任何全函数 f，

$f \leqslant_T \varnothing'$ 当且仅当 存在一个递归序列 $\{f_s\}_{s \in N}$ 使 $\lim_s f_s = f$。

证 设 $f \leqslant_T \varnothing'$。由于 $\varnothing'$ 是 $r.e$ 集，根据定理9.3.2，存在一个递归序列 $\{f_s\}_{s \in N}$，使

$$\lim_s f_s = f。$$

反之，设有递归序列 $\{f_s\}_{s \in N}$，使

$$\lim_s f_s = f。$$

令

$$A_x = \{s \mid \exists t(s \leqslant t \wedge f_t(x) \neq f_{t+1}(x))\}。$$

由于序列 $\{f_s\}_{s \in N}$ 是收敛的，所以对每个 x，A_x 是有穷集。令

$$B = \bigoplus A_x =_{df} \{(s,x) \mid s \in A_x\}。$$

易见

$$(s,x)\in B \Leftrightarrow s\in A_x \Leftrightarrow \exists t(s\leqslant t \wedge f_t(x)\neq f_{t+1}(x))。$$

由于$\{f_s\}_{s\in N}$是递归序列，所以 $f_t(x)\neq f_{t+1}(x)$是递归谓词。而 $s\leqslant t$ 当然是递归谓词，因此$\exists t(s\leqslant t \wedge f_t(x)\neq f_{t+1}(x))$是递归可枚举谓词，也就是说 B 是 $r.e$ 集(从而 $B\leqslant_T\varnothing'$)，$s\in A_x$ 是递归可枚举谓词。

给定 x，我们可以 B-递归(从而也是$\varnothing'$递归)地计算出最小模

$$m(x)=\mu s[s\notin A_x],$$

于是

$$f\leqslant_T m\leqslant_T B\leqslant_T\varnothing'。$$

定理9.3.3为我们提供了一个构造$\mathbf{0}'$以下的图灵度的方法：如果我们能构造一个0,1值的递归序列$\{f_s\}_{s\in N}$使之收敛于 A(将集合与其特征函数等同看待)，则 $\deg(A)\leqslant\mathbf{0}'$。

如果所构造的这个序列还是单调上升的，则 A 就是一个 $r.e$ 集，$\deg(A)$就是一个 $r.e$ 度。

在下一章实际构造一个 $r.e$ 度 $\deg(A)$时，我们的一般作法是构造一个上升的集合序列

$$A_0\subseteq A_1\subseteq\cdots\subseteq A_s\subseteq\cdots$$

使得 $x\in A_s$ 是(关于 x 和 s 的)二元递归谓词，然后令 $A=\bigcup_s A_s$。

当我们把 A_s 看成特征函数时，序列$\{A_s\}_{s\in N}$就是一个单调上升的函数序列，A 就是它的极限。根据定理9.3.1，它有最小模 $m\leqslant_T A$。

如果$\{A_s\}_{s\in N}$是一个上升的集合序列，$x\in A_s$ 是(关于 x 和 s 的)二元递归谓词并且 $A=\bigcup_s A_s$，我们就称$\{A_s\}_{s\in N}$是 A 的一个**递归的枚举序列**。不难看出，每个 $r.e$ 集都有递归的枚举序列，因为对每个 W_e，$\{W_{e,s}\}_{s\in N}$就是 W_e 的一个递归的枚举序列。

习　　题

1. 构造图灵机 Z，使在神谕 A 下 Z 所计算的函数是 $\overline{A}$(的特

征函数)，即

$$\Phi_Z^A(x)=1-C_A(x)。$$

2. 对集合 A，B，定义

$$A\oplus B=\{2x \mid x\in A\}\cup\{2x+1 \mid x\in B\}。$$

证明：$\deg(A\oplus B)$是 $\deg(A)$和 $\deg(B)$的上确界，即

(1) $A\leqslant_T A\oplus B$ 且 $B\leqslant_T A\oplus B$；

(2) 对任何 C，如果 $A\leqslant_T C$ 且 $B\leqslant_T C$，则 $A\oplus B\leqslant_T C$。

3. 证明：$\{(e,\sigma,x,y,s)\mid\{e\}_s^\sigma(x)=y\}$是递归集。

4. 证明：$\{(e,\sigma,x,y)\mid\{e\}^\sigma(x)=y\}$是 $r.e$ 集并且是 T-完全的。

5. 定义 $A^{(\omega)}=\{(x,y)\mid x\in A^{(y)}\}$。

(1) 证明：对任意的 n，$A^{(n)}\leqslant_T A^{(\omega)}$。

(2) 证明：存在递归全函数 h，使对所有的 n，$\Phi_{h(n)}^{A^{(\omega)}}=A^{(n)}$。

(3) 如果存在递归全函数 f，使对全体 n

$$\Phi_{f(n)}^B=A^{(n)},$$

则称 B 是全体 $A^{(n)}$的一致上界。

证明：如果 B 是 $A^{(n)}(n=0,1,2,\cdots)$的一致上界，则 $A^{(\omega)}\leqslant_T B$(即 $A^{(\omega)}$是 $A^{(n)}$的最小一致上界)。

第十章

波斯特问题

1944年，波斯特提出一个著名的问题：

*是否存在图灵度 **a** 使 $\mathbf{0}<\mathbf{a}<\mathbf{0}'$？*

这个问题被称为**波斯特问题**。在此后的50年中，由波斯特问题所引发的对 $r.e$ 度的研究一直是递归论的主要研究方向。由此而创造和发展起来的优先方法是数理逻辑中与力迫法齐名的重要方法。

§10.1 波斯特问题和单纯集

波斯特认为他所提出的问题的答案应当是肯定的，即存在图灵度 **a** 使 $\mathbf{0}<\mathbf{a}<\mathbf{0}'$。为此，他提出了单纯集的概念，并证明了单纯集的存在性(波斯特所构造的单纯集是一个能行单纯集)。由于单纯集不是 m-完全的，所以波斯特希望能证明单纯集也不是图灵完全的。但下面的定理却否定了波斯特的想法。

定理10.1.1 如果 A 是能行单纯集，则 $K\leqslant_T A$。

证 令 $\{K_s\}_s\in N$ 是 K 的一个递归的枚举序列。定义

$$\theta(x)=\begin{cases}\mu s[x\in K_s] & \text{如果 } x\in K\\ \uparrow & \text{如果 } x\notin K\end{cases}$$

易见，$\theta(x)$是递归函数。

设 A 是能行单纯集(参看定义8.3.4)，f 是递归全函数，使得

如果 $W_x\subseteq A$，则 $|W_x|<f(x)$。

令 $\{A_s\}_{s\in N}$是 A 的一个递归的枚举序列(比如当 $A=W_e$ 时，令

$A_s = W_{e,s}$)。记

$$\overline{A_s} = \{a_0^s < a_1^s < \cdots < a_i^s < \cdots\}。$$

其中 a_i^s 是$\overline{A_s}$中(依从小到大的次序)的第 i 个数。

注意 a_i^s 是 s 和 i 的递归全函数,因为$\{A_s\}_{s\in N}$是递归序列,所以我们可以按下述方法能行地计算 a_i^s:

$$a_0^s = \mu x[x \notin A_s],$$

$$a_1^s = \mu x[x > a_0^s \wedge x \notin A_s],$$

$$a_2^s = \mu x[x > a_1^s \wedge x \notin A_s],$$

…………

$$a_{i+1}^s = \mu x[x > a_i^s \wedge x \notin A_s],$$

…………

此外,由于 A 是单纯集,$\overline{A}$ 无穷,而 $\overline{A} \subseteq \overline{A_s}$,所以 a_i^s 是 s 和 i 的全函数。

记

$$\overline{A} = \{a_0 < a_1 < \cdots < a_i < \cdots\}。$$

a_i 是 $\overline{A}$ 中(依从小到大的次序)的第 i 个数。根据同样的道理、a_i 是 i 的 A 递归全函数。

此外,由于 $\overline{A} \subseteq \cdots \subseteq \overline{A_{s+1}} \subseteq \overline{A_s} \subseteq \cdots \subseteq \overline{A_0}$,对任意的 s 和 i,

(1) $a_i^s \leqslant a_i$,

(2) 如果 $a_i^s = a_i$,则对所有的 $k \leqslant i$, $a_k^s = a_k$。

定义三元函数 $g(x,y,u)$ 为

$$g(x,y,u) = \begin{cases} 1 & \text{当 } x \in K \wedge \exists t(t \leqslant y \wedge u = a_t^{\theta(x)}) \\ \uparrow & \text{否则。} \end{cases}$$

由于 $t \leqslant y \wedge u = a_t^{\theta(x)}$ 是递归可枚举谓词,加上存在量词$\exists t$ 后所得到的$\exists t(t \leqslant y \wedge u = a_t^{\theta(x)})$仍是递归可枚举谓词,而 $x \in K$ 也是递归可枚举谓词,所以 g 是三元递归函数。根据 s-m-n 定理,存在二元递归全函数$m(x,y)$使

$$\varphi_{m(x,y)}(u) = g(x,y,u).$$

用递归全函数 $f(y)$代入 y,得到

$$\varphi_{m(x,f(y))}(u)=\begin{cases}1 & \text{当 } x\in K\wedge\exists t(t\leqslant f(y)\wedge\ u=a_t^{\theta(x)})\\ \uparrow & \text{否则。}\end{cases}$$

于是得到

$$W_{m(x,f(y))}(u)=\begin{cases}\{a_0^{\theta(x)},a_1^{\theta(x)},\cdots,a_{f(y)}^{\theta(x)}\} & \text{当 } x\in K,\\ \varnothing & \text{否则。}\end{cases}$$

对递归全函数 $m(x,f(y))$ 使用带参数的递归定理，得到

$$W_{n(x)}=W_{m(x,f(n(x)))}=\begin{cases}\{a_0^{\theta(x)},a_1^{\theta(x)},\cdots,a_{f(n(x))}^{\theta(x)}\} & \text{当 } x\in K,\\ \varnothing & \text{否则。}\end{cases}$$

令

$$r(x)=\mu s[a^s_{f(n(x))}=a_{f(n(x))}]。$$

由于 $a^s_{f(n(x))}$ 是递归全函数而 $a_{f(n(x))}$ 是 A-递归全函数，所以

$$a^s_{f(n(x))}=a_{f(n(x))}$$

是 A-递归谓词，从而 $r\leqslant_T A$。

当 $x\in K$ 时，如果 $r(x)\leqslant\theta(x)$，则由 $r(x)$ 的定义 $a^{\theta(x)}_{f(n(x))}=a_{f(n(x))}$。于是根据(2)，对每个 $k\leqslant f(n(x))$，$a_k^{\theta(x)}=a_k$，从而

$$W_{n(x)}\subseteq\overline{A},$$

而此时

$$|W_{n(x)}|=f(n(x))+1,$$

与 f 的定义(即 A 的能行单纯性)矛盾。所以当 $x\in K$ 时 $r(x)>\theta(x)$。这样，由 r 和 θ 的定义就有

$$x\in K\Leftrightarrow x\in K_{\theta(x)}\subseteq K_{r(x)}\subseteq K,$$

也就是

$$x\in K\Leftrightarrow x\in K_{r(x)}。$$

于是 K 是 r-递归的，而 $r\leqslant_T A$，所以 $K\leqslant_T A$。

为了证明另一个与单纯集的度有关的定理(定理10.3)，我们先证明：

定理10.1.2　如果 $\{A_s\}_{s\in N}$ 和 $\{B_s\}_{s\in N}$ 分别是递归可枚举集 A 和 B 的递归的枚举序列，并且满足

$$\text{如果 } x\in A_{s+1}-A_s,\ \text{则}(\exists y<x)(y\in B-B_s),\qquad ①$$

则 $A\leqslant_T B$。

证 任给 x，可以 B-递归地计算出

$$B\cap[0,1,\cdots,x-1]。$$

由于 $\{B_s\}_{s\in N}$ 是递归序列，所以可以 B-递归地验证是否有

$$B\cap[0,1,\cdots,x-1]=B_s\cap[0,1,\cdots,x-1]$$

因此可以 B-递归地找到最小的 s，使

$$B\upharpoonright x=B_s\upharpoonright x。\quad ②$$

若 $x\in A$，则对某个 t，$x\in A_{t+1}-A_t$，由①知存在 $y<x$ 使 $y\in B-B_t$。如果 $t>s$，则由 $B_s\subseteq B_t$ 知 $y\in B-B_s$，这与②矛盾，所以 $t<s$。于是 $A_t\subseteq A_s$，因此 $x\in A_s$。反之，若 $x\in A_s$，当然有 $x\in A$。从而得到

$$x\in A \text{ 当且仅当 } x\in A_s。$$

所以 $A\leqslant_T B$。

定理10.1.3 对任何非递归的递归可枚举集 B，存在单纯集 A，使 $A\equiv_T B$。

证 由于 B 是非递归集，故 B 和 $\overline{B}$ 都是无穷集。设 f 是 B 的一个1-1的枚举函数，即 f 是1-1的递归全函数，$B=\text{ran } f$。

令 $\{B_s\}_{s\in N}$ 是 B 的一个递归的枚举序列，其中

$$B_s=\{f(0),f(1),\cdots,f(s-1)\}。$$

我们来构造一个递归的集合序列 $\{A_s\}_{s\in N}$，使 $A=\bigcup_s A_s$ 是定理所要求的集合，即

1. A 是单纯集，
2. $A\equiv_T B$。

构造分阶段归纳进行。

第0个阶段：令 $A_0=\varnothing$。

设第 s 阶段已完成，A_s 已构造好，而

$$\overline{A_s}=\{a_0^s<a_1^s<\cdots<a_k^s<\cdots\},$$

其中 a_i^s 是 $\overline{A_s}$ 中(依从小到大的次序)的第 i 个数，则 a_i^s 是 s 和 i 的递归全函数(参看定理10.1.1的证明)。

第 $s+1$ 阶段(在 A_s 的基础上增加若干元素构造 A_{s+1})，分两步

进行：

第一步（为保证 A 是单纯集和 $A\leqslant_T B$）

对每个 $e\leqslant s$，考察

$$W_{e,s}\cap A_s=\varnothing \quad ③$$

及

$$\exists x(x>3e\wedge x\in W_{e,s}\wedge f(s+1)<x)。\quad ④$$

如果对于某个 e，③和④都真，则将使上述④式为真的最小的 x 放入 A_{s+1}；如果对任何 $e\leqslant s$，③和④都不全为真，则这一步就什么都不做。

由于第一步是针对 $e\leqslant s$ 的，所以这一步至多会有 $s+1$ 个数放入 A_{s+1}。

第二步（为保证 $B\leqslant_T A$）

将 $a^s_{3f(s+1)+1}$ 放入 A_{s+1}。这一步恰将一个数放入 A_{s+1}

下面证明如此构造的集合满足1和2两项要求。

(1) 考虑闭区间 $[0, 3i]$。这个区间中的数至多只有 i 个被在各阶段的第一步放入 A，因为根据④的第一款，在第一步中放入 A 的数 x 要满足 $x>3e$，因此，在这个区间中我们只可能针对 $e=0, 1, \cdots, i-1$ 将满足③和④的数放入 A。而根据④的第二款，如果我们在第 $s+1$ 阶段的第一步针对 e 将 x 放入 A，则 $x\in W_{e,s}$，因此对任何的 $t>s$，由于 $W_{e,s}\subseteq W_{e,t}$，在此后的第 $t+1$ 阶段的第一步中③不会再为真，也就不会再针对 e 将什么数放入 A。也就是说，在整个构造过程中，对一个 e 至多放入一个数。所以区间 $[1, 3i]$ 中至多有 i 个数放入 A。

如果 $y\in[0, 3i]$ 是在第 $s+1$ 阶段的第二步放入 A 的，则 $y=a^s_{3f(s+1)+1}$。而对任何的 s 和 k 我们有 $a^s_k\geqslant k$（参看定理10.1.1的证明），于是有

$$3f(s+1)+1\leqslant a^s_{3f(s+1)+1}=y\leqslant 3i,$$

从而

$$f(s+1)<i。$$

这样，由 s 的任意性就得到在各个阶段的第二步放入 A 的数也至多只有 i 个。

因此 $|\overline{A}\cap[0, 3i]|\geqslant i$，由 i 的任意性知 $\overline{A}$ 是无穷集。

(2) 我们来证明：如果 W_e 无穷，则存在 s 使④为真，从而有一个 $x\in W_e$ 在 $t\leqslant s+1$ 阶段放入 A。

用反证。设对任意的 s，④都为假，即

$$\forall s\forall x(x>3e\wedge x\in W_{e,s}\rightarrow x\leqslant f(s+1)) \qquad ⑤$$

为真。由于 W_e 无穷，对任意的 y，都可以递归地找到 t, x，使

$$x>3e \quad 且 \quad y<x\in W_{e,t}。$$

于是由⑤知，对任意的 $s\geqslant t$，

$$y<x\leqslant f(s+1)，$$

从而

$$y\in B \text{ 当且仅当 } y\in B_t，$$

也就是说 B 是递归集，与定理条件中的 B 不是递归集矛盾。这就表明：如果 W_e 无穷，则存在 s 使④为真。因此，如果 $W_{e,s}\cap A_s=\varnothing$，则在第 $s+1$ 级的第一步中有一个 $x\in W_e$ 被放入 A_{s+1}。所以，只要 W_e 无穷，则

$$W_e\cap A\neq\varnothing。$$

综合(1)和(2)，就证明了 A 是单纯集。

(3) 设 $x\in A_{s+1}-A_s$。如果 x 是在第一步放入 A_{s+1} 的，则根据④的第三款，

$$x>f(s+1)\in B-B_s。$$

如果 x 是在第二步放入 A_{s+1} 的，则

$$x=a^s_{3f(s+1)+1}\geqslant 3f(s+1)+1>f(s+1)\in B-B_s。$$

总之，$\{A_s\}_{s\in N}$ 和 $\{B_s\}_{s\in N}$ 满足定理10.1.2的条件①，所以根据定理10.1.2，$A\leqslant_T B$。

(4) 记

$$\overline{A}=\{a_0<a_1<\cdots<a_i<\cdots\}，$$

a_i 是 $\overline{A}$ 中(依从小到大的次序)的第 i 个数。根据定理10.1.1的证

明中的同样道理，a_i 是 i 的 A-递归全函数。

对任意的 x，可以 A-递归地找到 s，使

$$a^s_{3x+1}=a_{3x+1}$$

（参看定理10.1.1的证明）。从而对任何 k，

$$a^{s+k}_{3x+1}=a_{3x+1}\text{。} \qquad ⑤$$

如果 $x\in B$，则对某个 t，$x\in B_{t+1}-B_t$，根据 A 的构造，

$$a^t_{3x+1}\in A_{t+1}\text{，}$$

从而

$$a^t_{3x+1}\neq a_{3x+1}\text{。} \qquad ⑥$$

由⑥和⑦可知 $t\leqslant s$，于是

$x\in B$ 当且仅当 $x\in B_s$。

因此，$B\leqslant_T A$。

定理10.1.3表明，任何非递归的 $r.e$ 度中都含有单纯集，因此仅靠单纯集的概念无法区分完全的图灵度和其他非递归的 $r.e$ 度（如果有这样的度的话）。也就是说，仅靠单纯集这一个概念不能解决波斯特问题。

§10.2 优先方法

波斯特问题吸引了大批的逻辑学家和数学家投入了对图灵度结构的研究，波斯特本人更是将他生命中最后十年的主要精力花在这个问题之上。对波斯特问题的研究极大地推动了四、五十年代递归论的发展，尤其是对递归可枚举集的研究更是硕果累累。经过十二年的努力，这个问题终于在波斯特逝世两年之后由穆契尼克(1956)和弗里德伯格(1957)各自独立地解决了。他们解决波斯特问题的方法称为**优先方法**。本节我们先来介绍一个比较典型的使用优先方法的的证明，以便讲清这种方法，然后再介绍穆契尼克和弗里德伯格本来的证明。

定义10.2.1　设有图灵度 $\mathbf{a}\leqslant\mathbf{0}'$，如果 $\mathbf{a}'=\mathbf{0}'$，则称 $\mathbf{a}$ 为**低度**；

如果 $\mathbf{a}'=\mathbf{0}''$，则称 $\mathbf{a}$ 为**高度**。

根据定义，$\mathbf{0}$是一个低度，$\mathbf{0}'$是一个高度。这两个极端的低度和高度并没有什么特别的价值，但如果我们能找到一个非$\mathbf{0}$的低度 $\mathbf{a}$，那就解决了波斯特问题：因为那样就会有

$$\mathbf{0}<\mathbf{a}<\mathbf{a}'=\mathbf{0}'。$$

事实上确实存在这样的度：

定理10.2.1 存在具有低度的单纯集。

在证明定理之前，先来分析一下有关的一些问题，并对优先方法作些说明。我们的任务是构造一个递归的集合序列$\{A_s\}_{s\in N}$，使$A=\bigcup_s A_s$满足：

(1) $\overline{A}$ 无穷；

(2) 如果 W_e 无穷，则 $W_e\cap A\neq\varnothing$；

(3) $A'\leqslant_T\varnothing'$。

其中(1)和(2)保证 A 是单纯集；(3)保证 A 具有低度，即 $\deg(A')=\mathbf{0}'$。

要满足(1)比较容易：只要我们在构造递归序列$\{A_s\}_{s\in N}$时注意每将一个数放入 A_{s+1}时总使至少一个数留在 $\overline{A}$ 中就可以了。在定理8.3.7和定理10.1.3的证明中已经这样做过了。

为了满足(2)和(3)，我们设立以下两组**需求**：对每个 e

$$P_e:\quad W_e\text{ 无穷}\rightarrow W_e\cap A\neq\varnothing,$$

$$N_e:\quad(\exists^\infty s)(\{e\}_s^{A_s}(e)\downarrow)\rightarrow\{e\}^A(e)\downarrow。$$

其中 P_e 称为**正需求**，N_e 称为负需求，$\exists^\infty s$ 的意思是"存在无穷多个 s"。容易看出，如果全部正需求都得到满足，则(2)成立。而全部的负需求如何能保证(3)则需要证明。

引理10.2.2 如果所有的 N_e 都得到满足，则 $A'\leqslant_T\varnothing'$。

证 定义二元函数 $g_s(e)$为：

$$g_s(e)=\begin{cases}1 & \text{如果}\{e\}_s^{A_s}(e)\downarrow,\\ 0 & \text{否则。}\end{cases}$$

易见 $g_s(e)$是二元递归函数。

设 N_e 为真，如果只有有穷多个 s 使$\{e\}_s^{A_s}(e)\downarrow$，则存在 t 使当 $s\geq t$时$\{e\}_s^{A_s}(e)\uparrow$，于是当 $s\geq t$ 时 $g_s(e)=0$。如果有无穷多个 s 使$\{e\}_s^{A_s}(e)\downarrow$，则由 N_e 知$\{e\}^A(e)\downarrow$。根据定理9.1.8，存在$\sigma\subset A$，使$\{e\}^\sigma(e)\downarrow$。由于 σ 有穷，所以存在 t 使得 $\sigma\subset A_t$，从而当 $s>t$ 时$\{e\}_s^{A_s}(e)\downarrow$。也就是说当 $s>t$ 时 $g_s(e)=1$。

因此，如果每个 N_e 都得到满足，则递归函数序列$\{g_s(e)\}_{s\in N}$有极限

$$g(e)=\lim_s g_s(e)。$$

而且

$$g(e)=1\ 当且仅当\ \{e\}^A(e)\downarrow,$$

即 $g(e)$是 A'的特征函数。由极限引理知 $g\leq_T\varnothing'$，也就是

$$A'\leq_T\varnothing'。$$

现在，我们所要做的事情就是给出一个构造$\{A_s\}_{s\in N}$的能行过程，使对每个 e，P_e 和 N_e 最终都能满足。要满足正需求 P_e 是比较简单的：只要在发现 $W_e\cap A_s=\varnothing$时，将满足 $x\in W_{e,s}\wedge x>2e$ 的 x 放一个到 A_{s+1}中就可以了。现在来看一下负需求的情形。

定义函数

$$r(e,s)=u(A_s;e,e,s),$$[①]

称为**抑制函数**(也可以形象地称为“墙”)。根据定义9.1.3，$r(e,s)$ 是二元的递归函数。

要使负需求 N_e 得到满足，就是要避免出现有无穷多个 s 使$\{e\}_s^{A_s}(e)\downarrow$，但$\{e\}^A(e)\uparrow$的情形。那么，在什么情况下会出现有无穷多个 s 使$\{e\}_s^{A_s}(e)\downarrow$但$\{e\}^A(e)\uparrow$这种事情呢?

回忆前面所说的：集合 A 是对一个上升的集合序列

$$A_0\subseteq A_1\subseteq\cdots\subseteq A_s\subseteq\cdots$$

取并集而得到的。在构造过程中要不断地向 A 中增加元素。随着 s 的增大，A_s 中的元素会逐渐增多，于是神谕带上的内容也就会不

---

① $u(A_s;e,x,s)$是使用函数，见定义9.1.3。

断地变化：如果在构造的第$s+1$步将x放入了A_{s+1}，神谕带上原来写有$A_s(x)=0$的方格中的内容就会变为$A_{s+1}(x)=1$。由于在$\{A_s\}_{s\in N}$的整个构造过程中要向A中放入无穷多个数，有神谕从0变为1的事情也就会发生无穷多次。因此，对任何一个固定的e而言，尽管图灵机e和输入e都是固定不变的，但神谕带上的神谕却在不停地变化，所以有可能出现随着s的增大$\{e\}_s^{A_s}(e)$忽而收敛忽而发散的情形。于是就可能出现有无穷多个s使$\{e\}_s^{A_s}(e)\downarrow$，但$\{e\}^A(e)\uparrow$的情形。例如当$A_s=\{0,1,2,3,\cdots,s\}$时，§9.1例2中的图灵机$Z$就是这样的情形。

当然，上面所说的情形只是可能出现，而不是必然出现(因此我们才去设法避免它)。如果$\{e\}_s^{A_s}(e)\downarrow$，则$r(e,s)>0$，就是说在神谕$A_s$下，向第$e$号图灵机输入$e$之后会形成一个计算，在该计算中所用到的神谕数小于$r(e,s)$。此后，如果不再向$A$中放入小于$r(e,s)$的数(即不在墙内增加新元素)，新增加的元素全都位于$r(e,s)$之外(墙外)，处在读写头达不到的位置上，因此也就对运算过程不会有任何影响，所以对任何$t>s$，$\{e\}_t^{A_t}(e)$的运算过程和$\{e\}_s^{A_s}(e)$的运算过程完全相同。也就是说，对每个$t>s$，$\{e\}_t^{A_t}(e)\downarrow$，从而$\{e\}^A(e)\downarrow$。如果能做到这一点，就可以使$N_e$从此得到满足。但是，如果为了满足$P_i$而将一个小于$r(e,s)$的数$x$放进了$A_{s+1}$(新元素放在墙内)，就有可能使$\{e\}_{s+1}^{A_{s+1}}(e)$的运算过程与$\{e\}_s^{A_s}(e)$不同，因为这次的改变位于读写头所能到达的某个位置上。这种现象称为为满足P_i而放入的x**损害**了N_e。因此，如果我们的构造过程能保证一旦$\{e\}_s^{A_s}(e)\downarrow$就不再将小于$r(e,s)$的数放入$A$，就能使需求$N_e$得到满足。

倘若我们所要对付的只是一个e，这件事并不难办到：只要要求第s步以后所加入A的数都大于$r(e,s)$就可以了。因为为了满足P_i，我们所要对付的只是无穷的W_i，既然W_i无穷，我们总能从中找到大于$r(e,s)$的元素。然而我们面对的却是无穷多个N_e，而且还要保持构造过程的能行性，事情就不那么简单了。我们无法要

求所要放入 A_{s+1} 的数同时大于

$$r(0,s),\ r(1,s),\ r(2,s),\ \cdots。$$

由上面的分析可以看出问题的症结在与正负需求之间可能存在冲突：为了满足正需求，我们可能要将一个数放入 A（所以叫做正需求）；而为了满足负需求，我们可能需要禁止某些数放入 A（所以叫做负需求）。正需求不会受到损害，因为正需求是要求将无穷的 W_e 中的一个元素放入 A，以作为 $W_e\cap A\neq\varnothing$ 的证据。这个证据一旦放入 A 就不会再取出来了。但正需求有可能损害负需求。根据上面的分析，要想回避这种损害是不现实的，优先方法就是为了解决这个矛盾而创造出来的。

优先方法的基本思想是给全部需求排定一个优先次序

$$N_0,\ P_0,\ N_1,\ P_1,\ N_2,\ P_2,\ \cdots,\ N_e,\ P_e,\ \cdots$$

在构造 $\{A_s\}_{s\in N}$ 的过程中，规定任何一个 P_i 都不准损害优先于它的 $N_e(e=0,1,\cdots i)$。具体做法是：当在第 $s+1$ 步 P_i 要求向 A 中放入一个数时，我们就放入一个大于

$$r(0,s),\ r(1,s),\ \cdots,r(i,s)$$

的数。因为只有有穷多个（$i+1$ 个）N_e 优先于 P_i，这个规定是可以做到的。这样，每个 N_e 都只会被它前面的 P_i 损害，而不会被它后面的 P_i 损害。由于每个 N_e 前面只有有穷多个 P_i，因此存在一个 s，使得到 s 步时 N_e 之前的 P_i 都得到满足，此后 N_e 就再也不会受到损害了。

定理10.2.1的证明　设立两组需求：对每个 e

$$P_e:\quad W_e\text{ 无穷}\rightarrow W_e\cap A\neq\varnothing,$$

$$N_e:\quad (\exists^\infty s)(\{e\}_s^{A_s}(e)\downarrow)\rightarrow\{e\}^A(e)\downarrow。$$

构造递归的集合序列 $\{A_s\}_{s\in N}$：

第0步，令 $A_0=\varnothing$。

第 $s+1$ 步，设 A_s 已构造好，$r(e,s)=u(A_s;e,e,s)$，看是否有 $i\leqslant s$ 满足下面两个条件：

$$W_{i,s}\cap A_s=\varnothing, \qquad ①$$

$$\exists x(x\in W_{i,s}\wedge>2i\wedge(\forall e\leqslant i)(r(e,s)<x))。\qquad ②$$

如果有这样的 i 存在，取其最小的一个，不妨就记作 i，将对于这个 i 满足②的最小的 x 放入 A_{s+1}。此时

$$W_{i,s}\cap A_{s+1}\neq\varnothing,\qquad ③$$

P_i 得到满足，我们说这一步关照了 P_i。

如果没有这样的 i 存在，则令 $A_{s+1}=A_s$。

令 $A=\bigcup_s A_s$。以下证明 A 符合定理的要求。

1. 如果 $x\in A_{s+1}-A_s$ 并且 $x\leqslant r(e,s)$ 就称 x 在第 $s+1$ 步损害了 N_e。定义对于 N_e 的**损害集** I_e 为：

$$I_e=\{x\mid\exists s(x\in A_{s+1}-A_s 且\ x\leqslant r(e,s))\}。$$

由①和③可以看出每个 P_i 只需要关照一次，而由②可以看出只有为了关照那些满足 $i<e$ 的 P_i 才可能损害 N_e，因此

$$|I_e|\leqslant e。$$

因为至多只有 e 个 x 会损害 N_e，所以存在 s_e 使得

$$如果\ x\in I_e\ 则\ x\in A_{s_e},\qquad ④$$

因此，如果对某个 $s>s_e$，$\{e\}_s^{A_s}(e)\downarrow$，则由④知对任何 $t\geqslant s$

$$\{e\}_t^{A_t}(e)\downarrow\wedge r(e,s)=r(e,t);\qquad ⑤$$

从而

$$\{e\}^A(e)\downarrow\wedge u(A;e,e)=r(e,s)。$$

如果对所有的 $s>s_e$，$\{e\}_s^{A_s}(e)\uparrow$，则对所有的 $s>s_e$，

$$r(e,s)=0。\qquad ⑥$$

由⑤和⑥知，极限 $r(e)=\lim_s r(e,s)$ 存在。

如果 $r(e)=0$，则至多有有穷多个 s 能使得 $\{e\}_s^{A_s}(e)\downarrow$；如果 $r(e)>0$，则 $\{e\}^A(e)\downarrow$。所以每个 N_e 都会得到满足。由引理10.2.2，

$$A'\leqslant_T\varnothing'。$$

2. 如果 W_i 是无穷的，我们来证明 $W_i\cap A\neq\varnothing$。

取 s 使得

$$(\forall t\geqslant s)(\forall e\leqslant i)(r(e,s)=r(e))。\qquad ⑦$$

由于极限 $r(e)=\lim_s r(e,s)$ 存在，这样的 s 总是有的[①]。

取 $s'>s$ 使得对任何 $j<i$，P_j 在 s' 步之后都不再需要关照。由于只有有穷多个 $j<i$，因此这样的 s' 也是有的。

取 $t>s'$ 使得

$$\exists x(x\in W_{i,t}\wedge x>2i\wedge(\forall e\leqslant i)(r(e)<x))。 \quad ⑧$$

由于 W_i 无穷，这样的 x 也肯定是存在的。

在构造 A 的第 $t+1$ 步时，如果仍然是 $W_{i,t}\cap A_t=\varnothing$，则按照构造的要求在这一步就应当将满足⑧的最小的 x 放入 A_{t+1} 了。从而 P_i 至多到第 $t+1$ 步就可以得到满足了。由 i 的任意性知每个 P_i 都可以得到满足。

3. 根据②中的第二款，对任何 i，区间 $[0, 2i]$ 中至多有 i 个数被放入 A，因此 $|\overline{A}\cap[0, 2i]|>i$，由 i 的任意性知 $\overline{A}$ 无穷。

综合2和3就得到 A 是单纯集。

推论10.2.3 存在图灵度 $\mathbf{a}$ 使 $\mathbf{0}<\mathbf{a}<\mathbf{0}'$。

证 令 A 如定理10.2.1，$\mathbf{a}=\deg(A)$。由于 A 是单纯集，所以 A 不是递归集，$\mathbf{0}<\mathbf{a}=\deg(A)$。而由定理**10.2.1**，$\mathbf{a}'=\deg(A')=\mathbf{0}'$。

§10.3 穆契尼克和弗里德伯格的证明

定理10.3.1 存在 $r.e$ 集 A 和 B，使 $A\not\leqslant_T B$ 且 $B\not\leqslant_T A$。

定理要求给出两个递归的集合序列 $\{A_s\}_{s\in N}$，$\{B_s\}_{s\in N}$，使得 $r.e$ 集 $A=\bigcup_s A_s$ 和 $r.e$ 集 $B=\bigcup_s B_s$ 满足：对任何 e，

$$\{e\}^B\neq A \quad 及 \quad \{e\}^A\neq B。$$

为此，设立两组需求：对每个 e

$$R_{2e}: \qquad \{e\}^B\neq A,$$

---

① 当然，由于 $r(e)$ 未必是递归函数，我们并不能保证递归地找到这样的 s，但在这里我们只需要用到 s 的存在性。下面的 s' 和 t 也是这种情形

R_{2e+1}:　　$\{e\}^A \neq B$。

我们先来分析一下如何才能使 R_{2e} 得到满足。由于 A 是特征函数，所以是全函数。要使 R_{2e} 得到满足，我们只要有一个 x，使

$$A(x)=1 \quad \text{但} \quad \{e\}^B(x)\uparrow \text{或} \{e\}^B(x)\downarrow=0$$

就可以了。这样的 x 称为 R_{2e} 的**证据**，记作 x_{2e}。

为此，当我们构造到第 s 步后，如果发现

$$\{e\}_s^{B_s}(x)\downarrow=0,$$

我们就将这个 x 取作 x_{2e} 的候选对象，将它放入 A_{s+1}，就可以使

$$\{e\}_s^{B_s}(x)\neq A_{s+1}(x)=A(x)。$$

令

$$r=r(2e,s+1)=u(B_s;e,x,s)。$$

如果此后不再将小于 r 的数放入 B，就能保证

$$B\upharpoonright r=B_s\upharpoonright r,$$

就可以使

$$\{e\}^B(x)=\{e\}_s^{B_s}(x)=0\neq A(x)=1,$$

从而使 R_{2e} 得到满足。

关于 r_{2e+1} 的情形是类似的，只是在上面的分析中将 A，B 互换，将 $2e$ 换成 $2e+1$ 就可以了。

与定理10.2.1相比，容易看出：R_{2e} 在构造 $\{A_s\}_{s\in N}$ 时是个正需求，它要求将一个证据放入 A；而在构造 $\{B_s\}_{s\in N}$ 时则是一个负需求，它要求禁止小于 $r(2e,s+1)$ 的数放入 B。相应地，R_{2e+1} 对于构造 $\{A_s\}_{s\in N}$ 是负需求，而在构造 $\{B_s\}_{s\in N}$ 时是正需求。

现在我们面临着三个问题。

第一个问题是在证明定理10.2.1时就遇到过的：由于负需求有无穷多个，我们在为满足一个正需求而将一个数放入 A(或 B)时，不可能保证不损害所有的负需求。解决这一问题的办法仍是象定理10.2.1的证明那样，规定各需求之间的优先次序——就依其下标的次序——要求在满足一个正需求时不得损害优先于它的(有穷多个)负需求。

第二个问题是：我们所要构造的 A 和 B 都不是递归的，因而都是无穷集。但是，同一个 x 却可以同时对无穷个 e 使

$$\{e\}^B(x) \neq A(x),$$

因此对无穷多个 R_{2e} 我们所放入 A 的证据可能是相同的。为了便于证明 A 是无穷集，我们利用配对函数将原本位于一根数轴上的自然数集

$$N=\{0, 1, 2, \cdots\}$$

对应于平面直角坐标系上位于一个象限上的二元数组集

$$\{(x,y)|x\in N, y\in N\},$$

如图10.1所示：

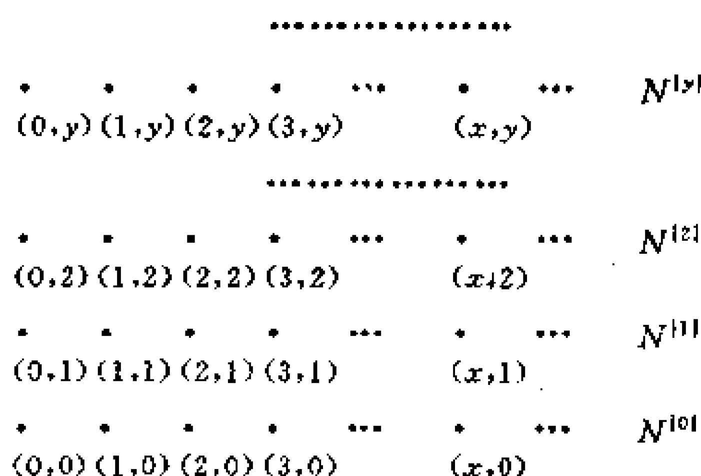


图10.1　候选证据

将其中(从下向上数)的第 e 行记作

$$N^{(e)}=\{(x,y)|y=e\}$$

(以下将 (x,y) 与 $J(x,y)$ 等同看待)。在构造过程中，为满足 R_e，我们只在 $N^{(e)}$ 中选取证据。这样就可以保证 A,B 都是无穷的。

第三，由于每个 R_e 都既是正需求又是负需求，所以就不象定理10.2.1那样对 R_e 放入一个证据就永远保证 R_e 得到满足。例如，在第 $t+1$ 步时发现

$$\{e\}_t^{B_t}(x)\downarrow=0,$$

将 x 作为满足 R_{2e} 的证据放入 A_{t+1} 而使

$$\{e\}_t^{B_t}(x)\downarrow=0\neq A_{t+1}(x)=1。$$

但到第 $s+1(s>t)$ 步时，如果为了满足某个优先于 R_{2e} 的 R_{2i+1} 而将某个数放入了 $B_{s+1}\supset B_{t+1}$，从而损害了 R_{2e}，即使得

$$\{e\}_{s+1}^{B_{s+1}}(x)\downarrow=1=A_{s+1}(x),$$

那么先前的证据 x 失去了效用。我们又需要另为 R_{2e} 准备一个候选的证据对象，而且这个候选证据还不得损害优先于 R_{2e} 的需求。为此我们需要标出哪些证据依然有效，哪些证据已经可能失效。由于我们无法知道可能失效的那些证据（无穷多个）中究竟有多少已经失效，因此干脆废弃所有的可能失效的证据。就是说，如果我们在第 $s+1$ 步时为满足需求 R_e 而将一个证据放入 A（或 B），那么对于全体 $j<e$，R_j 不会受到损害，于是此前为满足 R_j 而放入 B（或 A）的证据依然有效，而对全体 $j>e$，此前为满足 R_j 所放入 B（或 A）的证据就不算数了，因为它们可能已经失效。

定理的证明　递归地构造集合序列 $\{A_s\}_{s\in N}$ 和 $\{B_s\}_{s\in N}$，使之满足以下两组需求：

$$R_{2e}:\quad \{e\}^B\neq A,$$

$$R_{2e+1}:\quad \{e\}^A\neq B。$$

第0步　令

$$A_0=B_0=\varnothing,$$

对任何 e，令

$$x_e^0=(0,e),\qquad\text{（在0步后为 } R_e \text{ 准备的候选证据），}$$

及

$$r(e,0)=-1。$$

其中 x_e^0 是在0步后为 R_e 准备的候选证据，$r(e,0)$ 是0步后关于 R_e 的抑制函数。

第 $s+1$ 步　第每个 $e<s$，考察是否有

$$\{e\}_s^{B_s}(x_{2e}^s)\downarrow=0\wedge r(2e,s)=-1\qquad ①$$

或

$\{e\}_s^{A_s}(x_{2e+1}^s)\downarrow=0 \wedge r(2e+1,s)=-1$。 ②

如果有 $e<s$ 使①为真，则在这一步关照 R_{2e}：将 x_{2e}^s放入 A_{s+1}，并令

$x_{2e}^{s+1}=x_{2e}^s$ （将此证据记录在案）

及

$r(2e,s+1)=u(B_s;e,x_{2e}^s,s)$。 （立起一座墙）

然后，对每个 $j<2e$，令

$x_j^{s+1}=x_j^s$ （保留优先的证据）

及

$r(j,s+1)=r(j,s)$； （保留优先的墙）

对每个 $i>2e$，令

$r(i,s+1)=-1$ （推倒后面的墙）

及

$x_i^{s+1}=\mu y[y\in N^{[i]}\wedge y\notin(A_{s+1}\cup B_{s+1})\wedge y>\mathrm{Max}\{r(k,s+1)\mid k\leqslant 2e\}\wedge y>x_i^s]$ ③

（废除后面的证据，并确定新的候选证据）。

如果是②真，则情形类似，只是将上面过程中的 A 换成 B，B 换成 A，$2e$ 换成$2e+1$。如果①和②都为假，则维持原状。

最后，令 $A=\bigcup_s A_s$，$B=\bigcup_s B_s$，就完成了构造。

由于在这个构造中每个 R_e 都既是正需求也是负需求，因此除 R_0之外，每个 R_e 都可能受损害，每个 R_e 也都可能损害其后的需求。我们来分析一下 R_e 可能受损害的次数。记 I_e 为 R_e 的损害集。

R_0是最优先的需求，它不会受损害，因此$|I_0|=0$。

R_1只可能被 R_0损害，因此$|I_1|\leqslant 1$。

R_2受损害最多的情形是如下的次序：为关照（正需求）R_1而损害了 R_2（R_2作为负需求受损害一次），然后，R_2又得到关照。此后，为了关照（正需求）R_0而损害了 R_1和 R_2（R_2作为负需求第二次受损害），然后，R_2又得到关照。为了重新关照业已受损害的 R_1而再次损害 R_2（R_2第三次受损害）。因此，$|I_2|=3$。

利用归纳法可以证明：对任意的 e，$|I_e|\leqslant 2^e-1$。

下面我们来证明对这样构造出来的 A 和 B，每个需求 R_i 最终都会得到满足。用归纳法。先考虑 R_0。

根据构造过程，如果在第 $s+1$ 步（为关照优先于 R_i 的 R_j）损害了 R_i，则 x_i^s 会改变，也只有在这种情形 x_i^s 才会改变。由于 R_0 不会受到损害，因此对任何的 s，总有 $x_0^s=(0,0)$（记之为 x_0）。

如果对某个 s 有

$$\{0\}_s^{B_s}(x_0)\downarrow=0, \qquad ④$$

则根据构造，$x_0\in A$。由于 R_0 不受损害，所以

$$\{0\}^B(x_0)=\{0\}_s^{B_s}(x_0)=0\neq 1=A(x_0)。$$

如果对所有 s，④都不成立，则 $\{0\}^B(x_0)\uparrow$ 或 $\{0\}^B(x_0)\downarrow\neq 0$，而此时 $A(x_0)=0$。

总之 $\{0\}^B\neq A$。即 R_0 总会得到满足。

固定 i，设对任何 $j<i$，R_j 都满足。取定最小的 s，使在 s 步之后对任何 $j<i$，R_j 不再被关照。则对所有 $t\geqslant s$，x_i^t 不再变化，即

$$x_i^t=x_i^{t+1}（记之为 x_i）。$$

由构造过程中的③知 $x_i\notin A_s\cup B_s$。

不妨设 i 是偶数，$i=2e$（i 是奇数时情形类似，只需将 A，B 互换）。

如果对某个 $t\geqslant s$，R_{2e} 在 $t+1$ 步受到关照，则由①知

$$\{e\}_t^{B_t}(x_{2e})\downarrow=0\wedge x_{2e}\in A。$$

由于此后对任何 $j<2e$，R_j 都不再会受到关照，从而 R_{2e} 不会受到损害，因此

$$\{e\}^B(x_{2e})=\{e\}_t^{B_t}(x_{2e})=0\neq 1=A(x_{2e})。$$

而且，由于此后 R_{2e} 不再受损害，因此也不会再次受到关照。

如果在 s 步之后 R_{2e} 没有再受到关照，则 x_{2e} 也就不会放入 A，因此 $A(x_{2e})=0$，而由②知不会有 $\{e\}^B(x_{2e})=0$。

总之，$\{e\}^B\neq A$，即 R_{2e} 满足。

随着优先方法的创立，递归论进入了一个全新的时代。从此开始了对图灵度特别是对递归可枚举的图灵度的深入、细致的研究，得到了丰富的结果。

本书所介绍的定理10.2.1和定理10.3.1的证明中，为满足每个正需求只需将有穷多个数放入所构造的集合中，因此每个负需求只会被损害有穷多次。这样的优先方法称为**有穷损害的优先方法**。后来，萨克斯、拉克朗和申非尔德等人又创造了无穷损害的优先方法（参看[5]），证明了更多的定理。比如：

对任何 $r.e$ 度 $\mathbf{a}>\mathbf{0}$，存在 $r.e$ 度 $\mathbf{b}$ 使 $\mathbf{a}\not\leqslant\mathbf{b}$ 且 $\mathbf{b}\not\leqslant\mathbf{a}$。

对任何 $r.e$ 度 $\mathbf{a},\mathbf{b}$，如果 $\mathbf{a}<\mathbf{b}$，则存在 $r.e$ 度 $\mathbf{c}$ 使 $\mathbf{a}<\mathbf{c}<\mathbf{b}$（稠密性定理）。

存在 $r.e$ 度 $\mathbf{a},\mathbf{b}>\mathbf{0}$ 使得 $\mathbf{0}$ 是 $\mathbf{a},\mathbf{b}$ 的唯一的下界（极小对定理）。

存在 $r.e$ 度 $\mathbf{a},\mathbf{b}$，它们没有下确界。

存在非 $r.e$ 的度 $\mathbf{a}$，使 $\mathbf{a}<\mathbf{0}'$（这表明图9.2中的两个菱形不是重合的）。

存在极小度 $\mathbf{m}$（即 $\mathbf{m}>\mathbf{0}$ 且如果 $\mathbf{a}<\mathbf{m}$ 则 $\mathbf{a}=\mathbf{0}$）。

这些内容不可能在本书中一一介绍了，有兴趣的读者可参看[5]。

习　　题

1. 证明：对任何非递归的 $r.e$ 集 B，存在低的单纯集 A 使 $A\leqslant_T B$。

提示：取定 B 的一个递归的枚举序列 $\{B_s\}_{s\in N}$，仿照定理10.2.1 的证明构造递归的集合序列 $\{A_s\}_{s\in N}$，并要求 $\{A_s\}_{s\in N}$ 和 $\{B_s\}_{s\in N}$ 满足定理10.1.2的条件：

如果 $x\in A_{s+1}-A_s$，则 $(\exists y<x)(y\in B-B_s)$。

2. 证明：对任何非递归的 $r.e$ 集 C，存在 $r.e$ 集 A,B，使得

$A\leqslant_T C$ 且 $B\leqslant_T C$ 且 $A\not\leqslant_T B$ 且 $B\not\leqslant_T A$。

提示:取定 C 的一个递归的枚举序列$\{C_s\}_{s\in N}$,仿照定理10.3.1的证明构造$\{A_s\}_{s\in N}$和$\{B_s\}_{s\in N}$,并要求$\{A_s\}_{s\in N}$和$\{B_s\}_{s\in N}$都对$\{C_s\}_{s\in N}$满足定理10.1.2的条件,即

如果 $x\in A_{s+1}-A_s$,则$(\exists y<x)(y\in C-C_s)$。

如果 $x\in B_{s+1}-B_s$,则$(\exists y<x)(y\in C-C_s)$。

索 引

A

阿基米德辗转相除法 3

阿克曼函数 50

奥拉斯托散纳筛法 4

B

半可判定谓词 106

半可判定性 105

标准模型 83

波斯特问题 153

不动点定理 76

补全函数 8

部分递归函数 48

部分函数 7

C

产生函数 127

产生集 127

程序 12

初始的瞬时描述 12

创造集 130

D

带表达式 11

带参数的递归定理 77

带神谕的图灵机 134

单纯集 132

低度 159

第二递归定理 76

递归的枚举序列 151

递归定理 76

递归函数 48

递归集 50

递归可枚举的度 144

递归可枚举集 105

递归可枚举谓词 106

递归论 3

递归全函数 48

递归谓词 50

读写头 9

度 144

对角线方法 67
多一归约 116

F

范式 64
符号串 10
复合 30
复杂度 81
负需求 160

G

高度 160
哥德尔不完全性定理 96
哥德尔第二不完全性定理 100
哥德尔第一不完全性定理 100
哥德尔配数法 55
哥德尔语句 97
工作带 9
归约函数 116

H

函数 7

J

极限 148
极限引理 150
计算 14
简单迭代 42

K

可计算函数 16
克利尼范式 64
克利尼范式定理 63

L

赖斯定理 122
赖斯-沙皮罗定理 123

M

枚举函数 112
枚举序列 151
模 148
模引理 149

N

内部状态 10
能行单纯集 133
能行方法 3
能行过程 3
能行可计算 5
能行可判定 5

P

判定问题 103
配对函数 46

Q

丘奇论题 69
丘奇-图灵论题 69
取极小运算 33
全函数 7

S

神谕 135
神谕带 134
神谕数 139
使用函数 141
使用丘奇论题的证明 70
收敛 148
收敛模 148
输出 15
输出问题 102
输入 12
输入问题 103
数字可表示 87
瞬时描述 11
四元组 10
算法 3
算法可计算 5
算法可判定 5
孙子定理 90
损害 162
损害集 164

T

特征函数 7
跳跃 145
跳跃定理 146
跳跃集 145
跳跃算子 145
停机 14
停机问题 101
通用函数 64
通用图灵机 64
图灵等价 143
图灵度 144
图灵归约 143
图灵机 12
图灵可计算函数 16
图灵可计算集 38
图灵可判定谓词 37
图象定理 114

W

外延集 7

X

相对的带参数的递归定理 139
相对的递归定理 139
相对的 s-m-n 定理 139
形式语言 $\mathscr{L}$ 80

需求　160

Y

演绎定理　83
一致上界　152
抑制函数　161
优先方法　159
有界和　42
有界积　42
有界量词　45
有界取极小　43
有界 μ-运算　43
右 n 反转机　29
右 n 复制机　25
右 n 消去机　29
右 n+1减复制机　29
原始递归　32
原始递归函数　39
原始递归集　44
原始递归谓词　43
运算　13

Z

正需求　160
正则图灵机　21
指标集　121
指令　10
指令组　12
中国余数定理　90
終止的瞬时描述　14
状态　10
子函数　123
自对偶　118
字母　10
字母表　10
最小模　148
左 n 反转机　29
左 n 复制机　28
左 n 消去机　29
左 n+1增复制机　29

外文字母

A-递归函数　137
A-递归可枚举集　142
A-递归全函数　137
A-可计算函数　137
A-*r.e* 集　142
m-等价　119
m-度　120
m-归约　116
m-完全集　118
PA　79
PA 定理　83
PA 演绎　83
PA 证明　82
s-m-n 定理　72
r.e 度　144

r.e集　105
r.e谓词　106
μ-递归函数　48
μ-运算　33

参 考 书 目

[1] M. Davis *Computability and Unsolvability* 1958 中译本《可计算性与不可解性》(沈 泓 等译),北京大学出版社,1984

[2] S. C. Kleene *Introduction to Metamathematics* 1952 中译本《元数学导论》(莫绍揆 译),科学出版社,1984

[3] 张鸣华:《可计算性理论》,清华大学出版社;1984

[4] N. Cutland *Computability* Cambridge University Press, 1980

[5] R. I. Soare *Recursively Enumerable Sets and Degrees* Springer-Verlag, 1987